Approximation Algorithms for Geometric Optimization Problems

Sören Laue

Dissertation
Presented to the Faculties
of Natural Sciences and Technology
of Saarland University
in Candidacy for the Degree of
Doctor of Engineering Sciences (Dr.-Ing.)

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurswissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken
2008

Tag des Kolloquiums:	28. August 2008	
Dekan:	Prof. Dr. Joachim Weickert	
Prüfungsausschuss:	Prof. Dr. Reinhard Wilhelm	(Vorsitzender)
	Prof. Dr. Stefan Funke	(Berichterstatter)
	Prof. Dr. Kurt Mehlhorn	(Berichterstatter)
	Prof. Dr. Friedrich Eisenbrand	(Berichterstatter)
	Dr. Khaled Elbassioni	

Abstract

This thesis deals with a number of geometric optimization problems which are all NP-hard. The first problem we consider is the set cover problem for polytopes in $\mathbb{R}^3$. Here, we are given a set of points in $\mathbb{R}^3$ and a fixed set of translates of an arbitrary polytope. We would like to select a subset of the given polytopes such that each input point is covered by at least one polytope and the number of selected polytopes is minimal. By using *epsilon-nets*, we provide the first constant-factor approximation algorithm for this problem. The second set of problems that we consider are power assignment problems in wireless networks. Ad hoc wireless networks are a priori unstructured in a sense that they lack a predetermined interconnectivity. We consider a number of typical connectivity requirements and either give the first algorithms that compute a $(1+\epsilon)$-approximate energy efficient solution to them, or drastically improve upon existing algorithms in running time. The algorithms are based on *coresets*. We then extend the algorithms from the Euclidean case to metrics of *bounded-doubling* dimension and study metric spaces of bounded-doubling dimension more in-depth. The last problem that we consider is the k-hop minimum spanning tree, that is, we are given a graph and a specified root node and we would like to find a minimum spanning tree of the graph such that each root-leaf path contains at most k edges. We give the first PTAS for the problem in the Euclidean plane.

Kurzzusammenfassung

Diese Arbeit befasst sich mit geometrischen Optimierungsproblemen, die alle NP-schwer sind. Das erste Problem, das wir betrachten, ist das Set Cover Problem für Polytope im $\mathbb{R}^3$. Hierbei sind Punkte im $\mathbb{R}^3$ und eine Menge von nicht verschiebbaren Polytopen gegeben, die Translationen eines Polytopes sind. Ziel ist es, eine Teilmenge dieser Polytope so auszuwählen, dass jeder Eingabepunkt überdeckt ist und die Zahl der ausgewählten Polytope minimal ist. Durch das Bestimmen von kleinen *epsilon-nets* entwickeln wir den ersten Approximationsalgorithmus mit konstanter Approximationsgüte. Als nächstes betrachten wir Power Assignment Probleme in drahtlosen Netzwerken. Drahtlose ad hoc Netzwerke sind a priori strukturlos, d.h. sie besitzen keine vorbestimmten festen Verbindungen. Wir betrachten eine Zahl von typischen Konnektivitätsanforderungen und entwickeln Algorithmen, die entweder die ersten $(1+\epsilon)$-Approximationen liefern oder vorherige Algorithmen deutlich in Laufzeit unterbieten. Die Algorithmen basieren auf *coresets*. Danach erweitern wir diese Algorithmen vom Euklidischen Raum auf metrische Räume mit begrenzter *doubling-dimension* und untersuchen solche Räume genauer. Das letzte Problem dem sich diese Arbeit widmet, ist das k-hop Minimum Spanning Tree Problem. Hierbei ist ein Graph und ein Wurzelknoten gegeben und man möchte einen minimalen Spannbaum finden, so dass jeder Wurzel-Blatt-Pfad höchsten k Kanten hat. Für die Euklidische Ebene entwickeln wir den ersten PTAS für dieses Problem.

Acknowledgements

I could not have written this thesis without the help of many people. First of all, I thank my advisor Prof. Dr. Stefan Funke for his guidance and helpful advice throughout my doctoral study. I am also grateful to Prof. Dr. Kurt Mehlhorn. It has been a great pleasure and honor to be a member of the Algorithms and Complexity group at the Max-Planck-Institute für Informatik. I also want to thank Prof. Dr. Friedrich Eisenbrand for his support and his instant commitment to become a referee for this thesis.

During the last three years I had many fruitful discussions with almost every member of our group. In particular, I would like to thank my co-authors Dr. Domagoj Matijević and Rouven Naujoks.

I would also like to express my gratitude to my friends and colleagues who contributed to this thesis in many ways: Stefan Canzar, Kevin Chang, Daniel Dumitriu, Khaled Elbassioni, Sathish Govindarajan, Nabil H. Mustafa and Saurabh Ray.

Table of Contents

List of Figures xi

1 Geometric Set Cover and Hitting Sets for Polytopes in $\mathbb{R}^3$ 5

1.1 Small Epsilon-nets for Polytopes in $\mathbb{R}^3$ 9

1.1.1 From Polytopes in $\mathbb{R}^3$ to Non-piercing Objects in $\mathbb{R}^2$ . . . 9

1.1.2 Small Epsilon-nets for Non-piercing Objects in $\mathbb{R}^2$ 14

1.2 From Epsilon-Nets to Hitting Sets 20

1.3 From Hitting Set to Set Cover 22

1.4 Conclusions and Open Problems 23

2 Power Assignment Problems in Wireless Networks 25

2.1 Introduction 25

2.2 Bounded-hop Energy-efficient Broadcast in $\mathbb{R}^2$ 27

2.2.1 Coresets 29

2.3 Minimum-energy Broadcast with Few Senders 35

2.3.1 Preliminaries 36

2.3.2 A Naive Algorithm 36

2.3.3 Small Coresets of the Network Topology 37

2.3.4 Faster $O(1)$-Approximations 40

2.4 Bounded-hop Multicast 43

2.4.1 Algorithms 43

2.5 Energy-minimal Network Coverage 44

2.5.1 A Small Coreset for k-disk Cover 45

2.5.2 Algorithms 47

2.5.3 k-disk Cover with Few Outliers 48

2.6 Information Aggregation via Energy-minimal TSP Tours 50

2.6.1 Why Euclidean TSP Does Not Work 51

2.6.2 A 6-Approximation Algorithm 51

3 Low-dimensional Metric Spaces 55

3.1 Introduction 55

3.2 Properties of Low-dimensional Metrics 57

3.2.1 Metrics of Bounded Doubling Dimension 57

3.2.2 Hierarchical Fat Decompositions (HFD) 58

3.2.3 A Characterization of Metrics of Bounded Doubling Dimension 59

3.3 Well-separated Pair Decomposition 61

3.4 Optimizing Energy-efficiency in Low-dimensional Metrics 62

3.4.1 Energy-efficient k-hop Broadcast, k-set Broadcast and k-disk cover 62

3.4.2 Energy-efficient k-hop Paths 63

3.5 Computing HFDs in Shortest-path Metrics 64

3.5.1 A Near-linear Time Algorithm 64

3.5.2 Hierarchical Routing in Doubling Metrics 65

3.6 Conclusions 65

4 k-hop Minimum Spanning Trees in Euclidean Metrics 67

4.1 Introduction 67

4.2 Preliminaries 70

4.3 The Algorithm 72

4.4 Generalizations 74

4.5 Conclusions and Open Problems 75

Bibliography **77**

List of Figures

1.1 An instance of the set cover problem for polytopes in $\mathbb{R}^2$. The shaded polytopes form a set cover. 5

1.2 An instance of the hitting set problem for polytopes in $\mathbb{R}^2$. The hitting set is formed by the points that are marked by a cross. . 6

1.3 All eight possible subsets of the input points can be constructed using the disks that are shown. This proves that the VC-dimension of disks in the plane is at least three. 7

1.4 A subset of the input points that lie diagonal using a disk is constructed. However, it is never possible to construct a subset using a disk that only contains the other two diagonal points. . . 7

1.5 It is impossible to construct the subset of the input points using a disk that only contains the three points that form the convex hull and that does not contain the fourth point in the middle. . . 7

1.6 The input point that is marked by a cross forms an epsilon-net for the choice of $\epsilon = 3/5$, i.e. every subset that contains at least 3 points needs to be hit by one point from the epsilon-net. 8

1.7 The input points that are marked by a cross form an epsilon-net for the choice of $\epsilon = 2/5$, i.e. every subset that contains at least 2 points needs to be hit by one point from the epsilon-net. 8

1.8 A grid is placed onto the space such that any grid cell contains at most one vertex of the input polytope. 10

1.9 The cone C and its internal ray r. 10

1.10 The lower envelope with respect to cone C, the corresponding cones are drawn dotted. 11

1.11 The flattened lower envelope with respect to cone C, the lower envelope is drawn dotted. 11

1.12 The assignment of non-lower envelope points to lower envelope points. 12

1.13 The projection τ from the flattened lower envelope onto the plane $z = 0$. 13

1.14 A set of non-piercing objects. 14

1.15 A set of two piercing objects. 14

1.16 Two intersecting $\mathcal{D}$-Delaunay edges and their defining objects. . 15

1.17 The sets S_i and the convex hull $\text{conv}_C(S)$ with respect to cone C. The $\mathcal{D}$-Delaunay triangulation is drawn dotted. 18

1.18 The corridor R which is split into two sub-corridors and two tri-colored triangles. The corners of the sub-corridors are marked by crosses. 18

1.19 The case where D contains colored points of at least two colors. . 19

1.20 The case where D contains colored points of exactly one color. . 20

1.21 A set system with the subsets A, B and C. 22

1.22 A set of points and planar polytopes is depicted on the left with the corresponding dual on the right. The dotted arrows show the correspondence between the points in the primal and the polytopes in the dual and the dashed arrows show the correspondence with the polytopes in the primal and the points in the dual. . . . 23

2.1 One transmission operation from node p to its neighboring nodes. 26

2.2 A power range assignment and the induced communication graph. 26

2.3 The source node s broadcasts its message to all other nodes within one hop. 28

2.4 The source node s broadcasts its message to all other nodes using intermediate nodes. 28

2.5 Original range assignment before the first phase 31

2.6 Range assignment after the first phase 31

2.7 Proof illustration for the constant factor approximation algorithm. 41

2.8 An optimal energy-minimal tour for points on a slightly bent line. 51

2.9 Tree T and its children trees $T_1, T_2, \dots, T_k$. 51

3.1 A hierarchical fat decomposition (HFD) for a point set is drawn here schematically. All five different levels from level 0 to level 4 are displayed. Each cluster of each level is represented by an ellipse and each landmark of each cluster is marked by a cross. An HFD naturally defines a tree. 59

3.2 A metric with unbounded doubling dimension but with bounded degree HFD. 60

4.1 A 2-hop minimum spanning tree. All root-leaf paths contain at most 2 edges. 67

4.2 The bounding box and the dissection of the plane with a point set is displayed. The thick lines correspond to level 0 lines, the thin lines correspond to level 1 lines and the dotted lines are level 2 lines. 71

4.3 All nodes actually lie on top of each other and the edges pass through one portal. 73

4.4 All nodes actually lie on top of each other and the dotted lines have length 0. 73

Introduction

Many real-world optimization problems have a geometric component, which can be twofold. They can have a geometric interpretation, like LP-type problems, where for instance the geometric insight led to the discovery of the first polynomial time algorithm for solving linear programs, or the problem itself can have a geometric background, like for instance covering a set of points in the Euclidean plane by a minimal number of disks. In this thesis we will focus on the latter, i.e. on optimization problems that stem from geometry.

The underlying geometry can change the hardness and approximability of a problem drastically. For instance, the famous Traveling Salesman Problem is NP-hard. On arbitrary graphs, the problem is even hard to approximate within any given constant. However, in the metric case, i.e. if the graph satisfies the triangle inequality, there is a simple polynomial time algorithm with a 1.5-approximation guarantee known. If we further refine the underlying geometry, i.e. we allow only a restricted set of metric spaces, namely metrics with bounded doubling dimension, we are able to find a $(1+\epsilon)$-approximation in quasi-polynomial time. And finally, in Euclidean space, we are able to find a $(1+\epsilon)$-approximate solution in polynomial time.

The TSP example shows the important role the geometry has on the hardness of a problem. The more restricted the geometry is the more efficiently the problem can be approximated. And often in real-world applications we naturally deal with a Euclidean space or with metric spaces that are in some sense close to Euclidean spaces. This thesis falls into this realm.

All problems that are considered in this thesis are NP-hard. This rules out the existence of algorithms that could compute the exact solution to these problems in polynomial time. Hence, we have to lower our expectations and look for approximate solutions with certain approximation guarantees. We will demonstrate the broad variety of techniques of how to solve a geometric optimization problem approximately.

Geometric Set Cover in $\mathbb{R}^3$

The first chapter deals with the set cover problem. It is a basic and very fundamental problem and one of the famous NP-hard problems already discussed in Garey and Johnson [GJ79]. Whereas the general set cover problem is hard to approximate better than a logarithmic factor, we will show that for the geometric version we can do better. Suppose we have points in the three-dimensional Euclidean space along with a set of copies of a polytope. Our task is to select a minimal number of these polytopes such that all points are covered by at least one polytope. For this problem we give the first constant-factor approximation algorithm. The algorithm is based on *epsilon-nets* and extends a result from Clarkson and Varadarajan [CV05] from unit-cubes to arbitrary, not necessarily convex polytopes.

Wireless Communication and Low-dimensional Metric Spaces

The second chapter discusses several problems that occur in wireless communication and can be also seen as special geometric versions of the set cover problem. Wireless network technology has gained tremendous importance in recent years. While the spatial aspect was already of interest in the wired network world, it has far more influence on the design and operation of wireless networks. The power required to transmit information via radio waves is heavily correlated with the Euclidean distance of sender and receivers. In contrast to wired or cellular networks, ad hoc wireless networks a priori are unstructured in a sense that they lack a predetermined interconnectivity. An ad hoc wireless network is built from a set of radio stations, each of which consists of a receiver as well as a transmission unit. A radio station p can send a message by setting its transmission range $r(p)$ and then by starting the transmission process. All other radio stations at distance at most $r(p)$ from p will be able to receive p's message. We consider several problems that arise in wireless communication and provide approximation algorithms that solve these problems approximately in an energy-efficient way. By using *coresets* we are able to either drastically improve existing algorithms in running time or give the first polynomial time approximation schemes for them.

For analytical purposes it is very convenient to assume that all network nodes are placed in the Euclidean plane. Unfortunately, in real-world wireless network deployments, especially if not in the open field, the experienced metric space does not exactly correspond to a Euclidean space. Buildings, uneven terrain or interference might affect the transmission characteristics considerably. Nevertheless there is typically a strong correlation between the actual geographic distance and the required transmission power. One mean to capture the similarity of the experienced metric space to low-dimensional Euclidean spaces is the *doubling dimension*. In the third chapter we will consider the doubling dimension more in-depth and give a novel characterization of such metrics. We then show how our algorithms for wireless communication problems can be adapted to arbitrary metric spaces of bounded doubling dimension.

k-hop Minimum Spanning Trees

The last chapter of this thesis deals with the k-hop minimum spanning tree problem, that is, given a graph and a specified root node we would like to find a minimum spanning tree of the graph such that each root-leaf path contains at most k edges. Whereas the minimum spanning tree problem is easily solvable exactly in polynomial time the added hop restriction makes the problem NP-hard. Again, the geometry has an important influence on the hardness of the problem. The problem is even hard to approximate within any sub-logarithmic factor for arbitrary graphs, since the set cover problem is a special case of it. In the metric case an approximation algorithm is known with logarithmic approximation guarantee. For Euclidean spaces we provide the first $(1 + \epsilon)$-approximation algorithm that runs in quasi-polynomial time. It even runs in polynomial time for the Euclidean plane.

Sources

The results from chapter one have been published at STACS 2008 [Lau08]. Chapter two and three are based on papers that have been published at STACS 2007 [FL07], at DCOSS 2007 [FLN07] and at DCOSS 2008 [FLNL08]. The results in chapter four have been published in Information Processing Letters [LM08].

Chapter 1

Geometric Set Cover and Hitting Sets for Polytopes in $\mathbb{R}^3$

Introduction

Suppose we are given a set of n points P in $\mathbb{R}^3$ and a collection of polytopes $\mathcal{T}$ that are all translates of the same polytope T. We consider two problems in this chapter. The first is the set cover problem where we want to select a minimal number of polytopes from the collection $\mathcal{T}$ such that their union covers all input points P. The second problem that we consider is finding a hitting set for the set of polytopes $\mathcal{T}$, that is, we want to select a minimal number of points from the input points P such that every given polytope is hit by at least one point. See Figures 1.1 and 1.2 for an illustration.

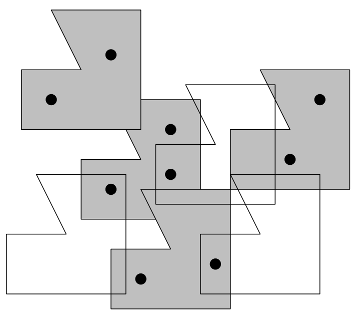

Figure 1.1: An instance of the set cover problem for polytopes in $\mathbb{R}^2$. The shaded polytopes form a set cover.

Both problems, the set cover problem and the hitting set problem, which are in fact dual to each other, are very fundamental problems and have been studied intensively. In a more general setting, where the sets could be arbitrary subsets,

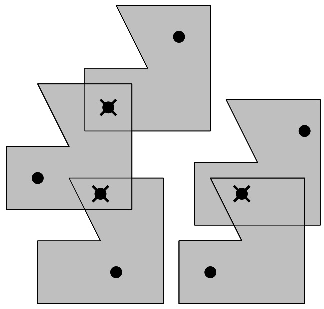

Figure 1.2: An instance of the hitting set problem for polytopes in $\mathbb{R}^2$. The hitting set is formed by the points that are marked by a cross.

both problems are known to be NP-hard. In fact they are even hard to approximate within any sub-logarithmic factor unless NP $\subseteq$ DITME($n^{\log \log n}$) [Fei98]. Even when the sets are induced by geometric objects it is widely believed that the corresponding set cover problem as well as the hitting set problem are NP-hard. Several geometric versions of these problems were even proven to be hard to approximate. Hence, we are looking for algorithms that approximate both problems. We give the first constant-factor approximation algorithms for the set cover problem and the hitting set problem for translates of a polytope in $\mathbb{R}^3$. The central concept of our approximation algorithms are small *epsilon-nets.*

A set of elements P (also called points) along with a collection $\mathcal{T}$ of subsets of P (also called ranges) is in general called a *set system* $(P, \mathcal{T})$ and for geometric settings also known as *range spaces*. One essential characteristic of these set systems is the *Vapnik-Chervonenkis dimension*, or *VC-dimension* [VC71]. The VC-dimension is the cardinality of the largest subset $A \subseteq P$ for which $\{T \cap A : T \in \mathcal{T}\}$ is the powerset of A. In this case we say the set A can be *shattered*. If the set A is finite, we say that the set system $(P, \mathcal{T})$ has bounded VC-dimension, otherwise we say the VC-dimension of $(P, \mathcal{T})$ is unbounded. For instance, the set system induced by disks in the plane has VC-dimension three as well as the set system induced by halfspaces in $\mathbb{R}^2$. Figure 1.3 illustrates the fact that the VC-dimension for disks in the plane is at least three. Three points can easily be shattered whereas there is no way to shatter four points by disks. The case where the four points lie in convex position is depicted in Figure 1.4. The other case where one point lies within the convex hull of the three others is depicted in Figure 1.5.

A set $N \subseteq P$ is called an *epsilon-net* for a given set system $(P, \mathcal{T})$ if $N \cap T \neq \emptyset$ for every subset $T \in \mathcal{T}$ for which $|T| \geq \epsilon \cdot |P|$. In other words, an epsilon-net is a hitting set for all subsets $T \in \mathcal{T}$ whose cardinality is at least an ϵ-fraction of the cardinality of the input point set P (cf. Figures 1.6 and 1.7).

We can generalize the definition of an epsilon-net to arbitrary measures in the obvious way: If we are given a measure $\mu : P \to \mathbb{R}_{\geq 0}$, a *weighted epsilon-net with respect to* μ is a set $N \subseteq P$ such that $N \cap T \neq \emptyset$ for every subset $T \in \mathcal{T}$ for

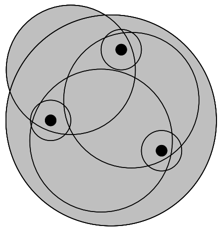

Figure 1.3: All eight possible subsets of the input points can be constructed using the disks that are shown. This proves that the VC-dimension of disks in the plane is at least three.

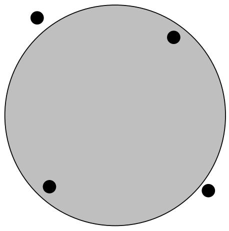

Figure 1.4: A subset of the input points that lie diagonal using a disk is constructed. However, it is never possible to construct a subset using a disk that only contains the other two diagonal points.

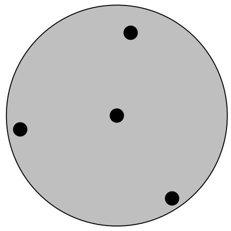

Figure 1.5: It is impossible to construct the subset of the input points using a disk that only contains the three points that form the convex hull and that does not contain the fourth point in the middle.

which $\mu(T) \geq \epsilon \cdot \mu(P)$, where $\mu(T) = \sum_{p \in T} \mu(p)$. Most proofs for epsilon-nets are done for unweighted epsilon-nets but they can be easily generalized to apply for weighted epsilon-nets as well.

Blumer et al. [BEHW89] showed that there exist epsilon-nets of size $O\left(\frac{d}{\epsilon}\log\frac{1}{\epsilon}\right)$ for any set system of VC-dimension d. This bound is in fact tight up to a multiplicative constant for arbitrary set systems as there exist set systems that do not admit epsilon-nets of size less than this bound [PW90]. Such an epsilon-net can be simply found by random sampling [Mat02]. The upper bound for the size of epsilon-nets was later improved to $(1 + o(1))\frac{d}{\epsilon}\log\frac{1}{\epsilon}$ by Komlós et al. [KPW92].

However, for special set systems that are induced by geometric objects there do exist epsilon-nets of smaller size, namely of size $O(\frac{1}{\epsilon})$. It has been shown by Pach and Woeginger [PW90] that halfspaces in $\mathbb{R}^2$ and translates of polytopes in $\mathbb{R}^2$ admit epsilon-nets of size $O(\frac{1}{\epsilon})$. Matoušek et al. [MSW90] gave an algorithm for computing small epsilon-nets for pseudo-disks in $\mathbb{R}^2$ and halfspaces in $\mathbb{R}^3$. The result for halfspaces in $\mathbb{R}^3$ also follows from a more general statement by Matoušek [Mat92].

Among other reasons for finding epsilon-nets of small size is the fact that

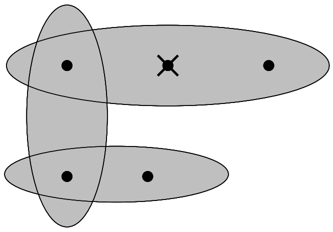

Figure 1.6: The input point that is marked by a cross forms an epsilon-net for the choice of $\epsilon = 3/5$, i.e. every subset that contains at least 3 points needs to be hit by one point from the epsilon-net.

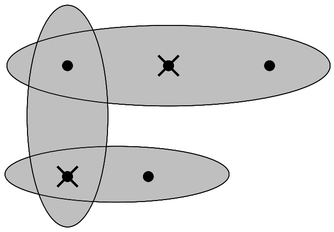

Figure 1.7: The input points that are marked by a cross form an epsilon-net for the choice of $\epsilon = 2/5$, i.e. every subset that contains at least 2 points needs to be hit by one point from the epsilon-net.

an epsilon-net of size $g(\epsilon)$ immediately implies an approximation algorithm for the corresponding hitting set problem with approximation guarantee of $O(g(1/c)/c)$, where c denotes the optimal solution to the hitting set problem [PA95]. This means, that for arbitrary set systems of fixed VC-dimension we have an algorithm for the hitting set problem with an approximation factor of $O(\log c)$. For set systems that admit epsilon-nets of size $O(1/\epsilon)$ we get an approximation algorithm to the hitting set problem with constant approximation guarantee.

Clarkson and Varadarajan [CV05] developed a technique that connects the complexity of a union of geometric objects to the size of the epsilon-net for the dual set system. Using this result, they are able to develop, among other approximation algorithms for geometric objects in $\mathbb{R}^2$, a constant-factor approximation algorithm for the set cover problem induced by translates of unit cubes in $\mathbb{R}^3$.

We extend their result to not only the set cover problem but also the hitting set problem for arbitrary translates of a polytope in $\mathbb{R}^3$. We do not require the polytope to be convex or fat. This is the first constant-factor approximation algorithm for these two problems. We achieve this by giving an epsilon-net for translates of a polytope in $\mathbb{R}^3$ of size $O(\frac{1}{\epsilon})$. We reduce the problem of finding epsilon-nets for translates of a polytope to a family of non-piercing objects in $\mathbb{R}^2$ and then generalize the epsilon-net finder for pseudo-disks of Matoušek et al. [MSW90] to our setting.

The set cover problem which is studied by Hochbaum and Maass [HM85] where one is allowed to move the objects is fundamentally different. They give a PTAS for their problem.

1.1 Small Epsilon-nets for Polytopes in $\mathbb{R}^3$

Let P be a set of n points in $\mathbb{R}^3$ and let $\mathcal{T}$ be a family of polytopes that are all translates of the same bounded polytope T_0. We want to find a set of polytopes of minimal cardinality among the collection $\mathcal{T}$ that covers all input points P. First, we find a small epsilon-net for this set system and use this later for a constant-factor approximation of the hitting set problem. Finally, we show how this can then be translated into a solution for the set cover problem.

We denote by T the polytope as well as the subset of points from P that T covers and by $\mathcal{T}$ the family of polytopes as well as the corresponding family of subsets of P. This will make this chapter easier to read and it will be clear from the context whether we refer to the geometric object or the corresponding set of points.

1.1.1 From Polytopes in $\mathbb{R}^3$ to Non-piercing Objects in $\mathbb{R}^2$

Given such a set system $(P, \mathcal{T})$ we want to find an epsilon-net for it, i.e. we are looking for a set $N \subseteq P$ such that every subset of points $T \in \mathcal{T}$ with $|T| \geq \epsilon \cdot |P|$ is stabbed by at least one point from N.

We can cut the polytope T into, lets say k polytopes $T_1, T_2, \ldots, T_k$. If the polytope T contains ϵn input points then one of the polytopes $T_1, T_2, \ldots, T_k$ must contain at least $\frac{\epsilon}{k} \cdot n$ input points. Hence, in order to find an ϵ-net for the set system $(P, \mathcal{T})$ induced by translates of T, it suffices to find an $\frac{\epsilon}{k}$-net for the set systems induced by the translates of $T_1, T_2, \ldots, T_k$.

Following this reasoning we can reduce our problem for finding an epsilon-net for the set system induced by translates of arbitrary polytopes to translates of *convex* polytopes by cutting the possibly non-convex polytope into a set of convex polytopes. Note that the number of these convex polytopes only depends on the polytope T and hence is constant for fixed T.

Without loss of generality let T be from now on a convex polytope. We can place a cubical grid onto the space $\mathbb{R}^3$ such that for any translate of T every cubical grid cell contains at most one vertex of T. This can be achieved by making the grid fine enough. Clearly, the maximal number t of grid cells that can be intersected by T is bounded and only depends on T. Again, if T contains $\epsilon \cdot n$ input points then at least one of the cells must contain at least $\frac{\epsilon}{t} \cdot n$ of the input points. Hence, we can restrict ourselves to finding epsilon-nets for translates of triangular cones where all input points lie in a cube in $\mathbb{R}^3$. This just adds a multiplicative constant to the size of the final epsilon-net. This idea is displayed for a two-dimensional example in Figure 1.8.

The case where the cubical cell only contains a halfspace or the intersection of two halfspaces can be either seen as a special case of a cone or, in fact, be even

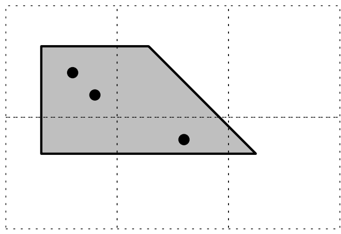

Figure 1.8: A grid is placed onto the space such that any grid cell contains at most one vertex of the input polytope.

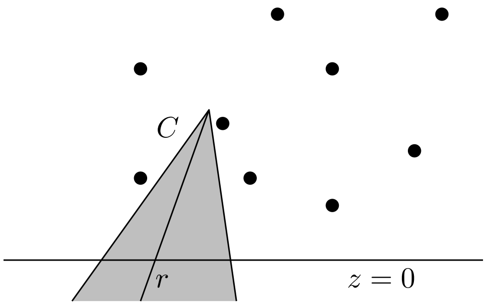

Figure 1.9: The cone C and its internal ray r.

treated separately in a much simpler way. The case of a translate of a halfspace reduces to a one-dimensional problem and admits an epsilon-net of size 1 and the case of two intersecting halfspaces reduces to a problem on intervals which admits an epsilon-net of size $O(1/\epsilon)$.

In the following we will construct an epsilon-net for the set system $(P, \mathcal{C})$ that is induced by translates of a triangular cone C.

Given a cone C, we call a set of points P in *non-C-degenerate position* if every translate of C has at most three points of P on its boundary. We can always perturb the input points P in such a way that they are in non-C-degenerate position and the collection of subsets of the form $P \cap C_T$ where C_T is a translate of C does not decrease [EW85]. Hence, we can restrict ourselves to non-C-degenerate sets of points P.

We place a coordinate system such that the input points all have z-coordinate greater than 0 and a ray r emitting from the apex of the cone C and lying entirely in the cone should intersect the plane $z = 0$. We refer to such a cone as a cone that *opens to the bottom* and the ray r as its *internal ray*. Figure 1.9 illustrates this setup for the two-dimensional case.

The following two definitions are helpful generalizations of the lower envelope.

Definition 1.1 *Given a finite point set P and a triangular cone C that opens* *to the bottom consider the arrangement of all translates of C that have a point of P on their boundary but no point of P in their interior. The upper set of plane segments that can be seen from above is called the* lower envelope of P with respect to cone C.

Figure 1.10 illustrates the definition of the lower envelope in the two-dimensional case. This definition is similar to the definition of alpha-shapes where the cone is replaced by a ball. We call all points that lie on the lower envelope with respect to cone C *lower envelope points* and denote this set by L.

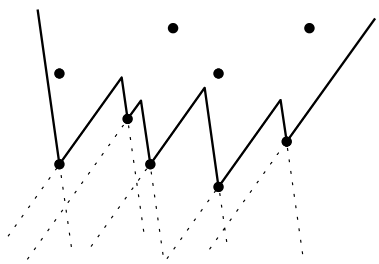

Figure 1.10: The lower envelope with respect to cone C, the corresponding cones are drawn dotted.

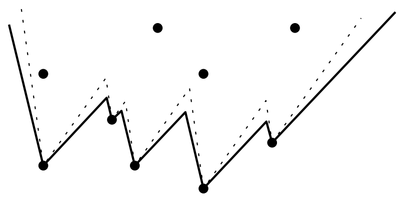

Figure 1.11: The flattened lower envelope with respect to cone C, the lower envelope is drawn dotted.

Definition 1.2 *Let C be a triangular cone that opens to the bottom and let $P \subseteq \mathbb{R}^3$ be a finite set of points in non-C-degenerate position. Let C' be a cone that is flatter than C by a small δ and such that it contains C and the combinatorial structure of P and C' is the same as for P and C. See Figure 1.11 for an illustration. Then, the lower envelope of P with respect to C' is called the* flattened lower envelope of P with respect to cone C.

Such a cone C' always exists for a finite point set that is in non-C-degenerate position. From now on we abbreviate the term lower envelope with respect to cone C by lower envelope since we will only deal with the same cone C. The flattened lower envelope can be basically seen as a slightly flattened version of the lower envelope.

The next lemma shows that we can reduce the problem of finding an epsilon-net with respect to cones of arbitrary point sets to finding a weighted epsilon-net of lower envelope points.

Lemma 1.3 *If for every finite point set $L \subseteq \mathbb{R}^3$ of lower envelope points in non-C-degenerate position there exists a weighted epsilon-net with respect to translates of a cone C of size $s(\epsilon)$ then there exists an epsilon-net with respect to translates of a cone C of size $s(\epsilon)$ for every finite point set $P \subseteq \mathbb{R}^3$ in non-C-degenerate position.*

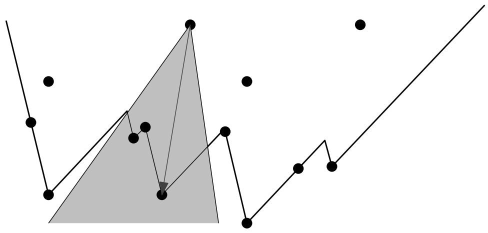

Figure 1.12: The assignment of non-lower envelope points to lower envelope points.

Proof: Let $P \subseteq \mathbb{R}^3$ be such a finite point set in non-C-degenerate position. Let L denote the set of lower envelope points, $\bar{L} = P \setminus L$ be the set of all non-lower envelope points and let $\mu : L \rightarrow \mathbb{R}_{\geq 0}$ be a measure for each point of the lower envelope. Initially, we set $\mu(p') = 1$ for each $p' \in L$. For each of the non-lower envelope points $p \in \bar{L}$ we do the following: We consider a cone C with apex p. Notice, that any cone that contains p must also contain C. Since p is a non-lower envelope point the cone C must contain a point $p' \in L$. We increase its measure $\mu(p')$ by 1. See Figure 1.12 for an illustration.

We claim that after doing so for all non-lower envelope points $p \in \bar{L}$ any weighted epsilon-net N' for L is also an epsilon-net for the set P. The following facts prove this statement.

1. The measure of all lower envelope points L is n, i.e. $\mu(L) = n$.

2. If an arbitrary cone C contains $\epsilon \cdot n$ points from P then the total measure μ of the points from L that C contains is at least $\epsilon \cdot n = \epsilon \cdot \mu(L)$.

Both properties show that the set N' is indeed an unweighted epsilon-net for P. □

The preceding lemma assures that we can restrict ourselves to a finite set of lower envelope points in non-C-degenerate position. For such a set system we will now construct a corresponding set system of points in the plane and a collection of regions in the plane.

Definition 1.4 *Let C be a cone and let P' be a finite set of lower envelope points in non-C-degenerate position and let $\mathcal{C}$ be a collection of translates of C. We define a projection τ from the flattened lower envelope onto the plane $z = 0$ by projecting each point along the internal ray r (cf. Figure 1.13). Let the projection of all points $p' \in P'$ which all lie on the plane $z = 0$ be denoted as the set S. For each cone of the collection the image of the intersection of the cone with the flattened lower envelope is an object $D \subseteq \mathbb{R}^2$ and the family $\mathcal{C}$ of cones induces a family of objects which we will denote by $\mathcal{D}$.*

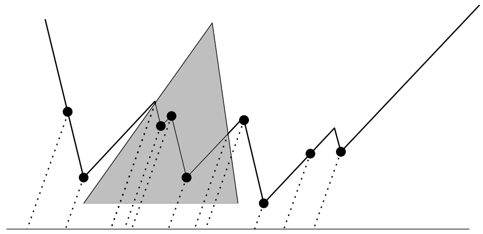

Figure 1.13: The projection τ from the flattened lower envelope onto the plane $z = 0$.

Using the flattened lower envelope instead of the lower envelope avoids degeneracy. The intersection of an arbitrary cone with the flattened lower envelope is always a collection of line segments. Furthermore, it makes everything continuous in the sense that if a cone is moved continuously in $\mathbb{R}^3$ then the intersection of the cone with the flattened lower envelope moves continuously as well as its image of the projection τ. Note that τ is injective.

Analogously, we call a set of points $S \subseteq \mathbb{R}^2$ in non-$\mathcal{D}$-degenerate position if every $D \in \mathcal{D}$ has at most three points on its boundary. We have the following lemma:

Lemma 1.5 *If for every finite point set $S \subseteq \mathbb{R}^2$ in non-$\mathcal{D}$-degenerate position there exists a weighted epsilon-net with respect to the family of objects $\mathcal{D}$ produced by the projection τ of size $s(\epsilon)$ then there exists a weighted epsilon-net with respect to cones of size $s(\epsilon)$ for every point set of lower envelope points $P' \subseteq \mathbb{R}^3$ in non-C-degenerate position.*

Proof: The proof follows immediately from the fact that the image of a cone C under the projection τ contains exactly those points that are the image of the points that are contained in C. $\square$

We refer to a cone C as the corresponding cone of the object $D = \tau(C)$. We will prove a few useful properties of the so constructed set system $(S, \mathcal{D})$.

Notice that the intersection of two triangular cones is again a cone. Furthermore, the intersection of a possibly infinite family of triangular cones is either empty or again a triangular cone since all cones are closed. The intersection of the boundary of a cone with the flattened lower envelope is either empty or a set of line segments that form one simple closed cycle. Hence, the image of a cone under the projection τ is a closed and connected region whose boundary is a closed and connected cycle.

We will now characterize the geometry of the family of objects $\mathcal{D}$.

Definition 1.6 *Two geometric objects(sets) $A \subseteq \mathbb{R}^2$ and $B \subseteq \mathbb{R}^2$ that are bounded by Jordan curves are said to be* non-piercing *if the boundary of A and*

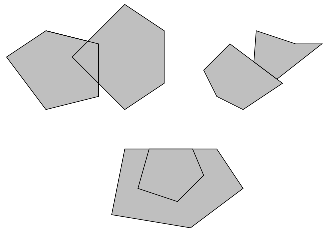

Figure 1.14: A set of non-piercing objects.

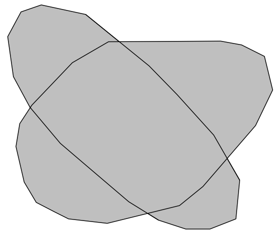

Figure 1.15: A set of two piercing objects.

the boundary B cross at most twice. A family of geometric objects is called non-piercing if every two objects from this family are non-piercing.

Figure 1.14 shows a set of non-piercing objects and Figure 1.15 shows two piercing objects for comparison.

Lemma 1.7 *The projection τ produces a family $\mathcal{D}$ of non-piercing objects.*

Proof: Consider two cones C_1 and C_2 that intersect each other. If one is contained in the other, i.e. $C_1 \subseteq C_2$, then we are done, as $\tau(C_1) \subseteq \tau(C_2)$ and hence their boundaries cannot cross. So if C_1 and C_2 intersect and none is subset of the other then the intersection of their boundaries are two rays emitting from the same point. Each of these rays intersects the flattened lower envelope exactly once. Hence, as the projection τ is injective, the boundary of the two images of the cones C_1 and C_2 under the projection τ intersect exactly twice. Thus, the objects are non-piercing. $\square$

1.1.2 Small Epsilon-nets for Non-piercing Objects in $\mathbb{R}^2$

In this subsection we will derive a few properties of the projection that are necessary to apply the algorithm of Matoušek et al. [MSW90] for finding a small epsilon-net for pseudo-disks. These properties also hold in general for any family of non-piercing objects with the additional property that for any three points there exists always an object that has these three points on its boundary. However, the proofs are a bit more involved. Since this does not lie in the scope of this chapter, we omit it here and focus only on the special family of non-piercing objects that is produced by the projection described above.

The idea behind the construction of the epsilon-net is to construct a planar graph with certain properties. If fact, the graph that we will construct can be seen as a generalization of a Delaunay-graph. We will show in a few technical lemmas that this generalization basically has the same properties as an ordinary Delaunay-graph, especially that it is a planar graph.

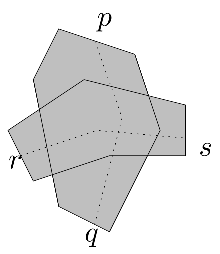

Figure 1.16: Two intersecting $\mathcal{D}$-Delaunay edges and their defining objects.

Consider the family of all cones that have p and q on its boundary. The intersection of all these cones is a cone C_{pq} that has p and q on its boundary. Connect p and q by a Jordan curve E_{pq} that lies entirely in the cone C_{pq} and on the flattened lower envelope, for instance part of the boundary of C_{pq} that intersects the flattened lower envelope. The image of E_{pq} under the projection τ is a curve $\tau(E_{pq})$ embedded in the plane.

Definition 1.8 *Let $\mathcal{D}$ be a family of non-piercing objects and let $S \subseteq \mathbb{R}^2$ be a finite set of points. We call two points $p, q \in \mathbb{R}^2$* $\mathcal{D}$-Delaunay neighbors *if there exists an object $D \in \mathcal{D}$ that has p and q on its boundary and no other point of S in its interior. The $\mathcal{D}$-Delaunay graph of S, in short $\mathcal{D}$-DT(S), is the graph that is embedded in the plane, has S as its vertex set and the edges $\tau(E_{pq})$ between all $\mathcal{D}$-Delaunay neighbors p and q.*

Due to the definition of the $\mathcal{D}$-Delaunay edge between two $\mathcal{D}$-Delaunay neighbors p and q it is guaranteed that whenever an object $D \in \mathcal{D}$ contains p as well as q then it also must contain the $\mathcal{D}$-Delaunay edge $\tau(E_{pq})$. In the following we will prove that this $\mathcal{D}$-Delaunay graph is in fact a triangulation of the vertex set S.

Lemma 1.9 *The $\mathcal{D}$-Delaunay graph of the given finite point set S in non-$\mathcal{D}$-degenerate position is a triangulation.*

Proof: First, we will prove that $\mathcal{D}$-DT(S) is planar. Suppose otherwise, i.e. two edges $\tau(E_{pq})$ and $\tau(E_{rs})$ intersect each other in the plane. Since the cone C_{pq} does not have any point in its interior and C_{rs} also does not have any point in its interior and since each of these cones has at most 3 points on its boundary the objects $\tau(C_{pq})$ and $\tau(C_{rs})$ would have to pierce each other, see Figure 1.16 for an illustration. Here it is actually essential that the set S is in non-$\mathcal{D}$-degenerate position. Thus, the graph is planar.

The graph $\mathcal{D}$-DT(S) itself consists of an outer face which is defined by cones of the lower envelope that have at most 2 points on their boundary and all other faces are triangles defined by the cones of the lower envelope that have exactly three points on its boundary. Suppose an inner face F is not bounded by a triangle. Then, one can place the apex of a cone in such a way onto the flattened lower envelope such that its image under the projection τ is a point which lies inside this face F. By moving the cone upward one can ensure that the cone will finally have three points on its boundary whose image under the projection τ are three vertices of the face F but no point in its interior. Hence, the face F must be bounded by a triangle. Hence, $\mathcal{D}$-DT(S) is a triangulation of the set S. □

We call the points of S that define the outer face the *convex hull of S with respect to cone C* and we denote it by $\text{conv}_C(S)$. It is a generalization of the standard convex hull and we will make use of it later. For a standard triangulation one requires that the outer face is determined by the convex hull. Here, we replaced the standard convex hull by the convex hull with respect to cone C. This is the appropriate generalization that we need.

Lemma 1.10 *Let D be an object produced by the projection τ. The subgraph G of $\mathcal{D}$-DT(S) induced by the points of S that lie in D is connected.*

Proof: We prove the connectivity using induction over the number of points that lie in D. If D contains at most 2 points then it must be connected by definition and by the fact that we can slide down the corresponding cone until both points lie on the boundary. So lets assume that every object D that contains at most k points from the set S induces a connected subgraph G. Now consider an object D that contains $k+1$ points of S. Consider the cone that is the intersection of all cones that contain exactly those $k+1$ points. This cone has exactly three points on its boundary. We can move the cone by a small δ in such a way that each of the three points can be excluded separately. As all of these induced graphs are connected by induction hypothesis, the whole subgraph induced by D must be connected. □

We need two more lemmas. Both lemmas basically rely on the fact that the projection τ is continuous.

Lemma 1.11 *Let S be a finite point set.*

1. *For any object $D \in \mathcal{D}$, there exists an object $D' \in \mathcal{D}$ with $S \cap D' = S \cap \text{int}\, D' = S \cap D$.*
2. *For any object $D' \in \mathcal{D}$, there exists an object $D \in \mathcal{D}$ with $S \cap D' = S \cap \text{int}\, D' = S \cap \text{int}\, D$.*

Proof: Let C be the corresponding cone of D. If we move C upward along the internal ray r by a small δ then the corresponding object D' of this cone will satisfy (1). On the other hand, if we move the cone C downward along the ray r by a small δ then the corresponding object D' will satisfy (2). □

Lemma 1.12 *Let S be a finite point set in non-$\mathcal{D}$-degenerate position, let (p, q) be a $\mathcal{D}$-Delaunay edge in $\mathcal{D}$-DT(S). Then there exists an object D with p and q on its boundary and with $S \cap D = \{p, q\}$.*

Proof: Let D be the object that assures that p, q is a $\mathcal{D}$-Delaunay edge, i.e. D has p and q on its boundary. Since the point set S is in non-$\mathcal{D}$-degenerate position D has at most three points on its boundary. If D has exactly two points on its boundary we are done. So lets assume that D has exactly three points on its boundary. Let C be the corresponding cone of D and let the corresponding points of p and q be $p' \in \mathbb{R}^3$ and $q' \in \mathbb{R}^3$. Neither p' nor q' can lie on the intersection of two of the defining planes of cone C because otherwise the cone could still be moved in an upward direction such that all three points still lie on the boundary until the cone hits a fourth point. But this would mean that the point set was in $\mathcal{C}$-degenerate position. Hence, p' and q' lie in the interior of two of the plane segments of cone C. If we now move the cone C downward by a small δ such that it still touches p' and q' then the corresponding object of this cone will only have p and q on its boundary. □

Having shown these properties, we can basically directly apply the algorithm for finding an epsilon-net for pseudo-disks from [MSW90]. We will describe the algorithm here and prove its correctness for our setting.

We are given a finite point set S in non-$\mathcal{D}$-degenerate position and we want to find a subset $N \subseteq S$ of size $O(1/\epsilon)$ that stabs any object D whose measure $\mu(D)$ is at least $\epsilon \cdot \mu(S)$.

Let $\delta = \epsilon/6$. First, let $S_1, \ldots, S_j$ be pairwise disjoint subsets of S with the following properties: Each S_i has a measure $\mu(S_i)$ of $\delta \cdot \mu(S)$, their union contains the convex hull of S with respect to cone C, i.e. $\text{conv}_C(S) \subseteq \bigcup_{1 \le i \le j} S_i$ and each S_i is representable by $S \cap \tau(C_i)$ for an appropriate cone C_i. Such sets can be easily constructed by repeatedly biting off points from $\text{conv}_C(S)$ with a suitable cone C_i. Notice that all these objects $D_i = \tau(C_i)$ belong to the collection $\mathcal{D}$.

Next, find a maximal pairwise disjoint collection $S_{j+1}, \ldots, S_k$ of subsets of the remaining points $S \setminus \bigcup_{1 \le i \le j} S_i$ satisfying $S_i = S \cap D_i$ for some object D_i and each subset has a measure of $\delta \cdot \mu(S)$. Obviously, there are at most $1/\delta + 1$ many subsets S_i in total. For an illustration we refer to Figure 1.17.

We assign all points in S_i the color i and call all other points *colorless*. Let $\bar{S}$ be the set of all colored points. Note that if an object contains only colorless points then its measure μ has to be less that $\delta \cdot \mu(S)$, since the collection of subsets S_i was maximal.

Let G be the $\mathcal{D}$-Delaunay graph of the set of colored points $\bar{S}$, i.e. $G = \text{DT}(\bar{S})$. G is indeed a triangulation (cf. Lemma 1.9). In this graph we call a triangle *uni-colored*, *bi-colored* or *tri-colored* depending upon the number of colors its vertices have. In a similar way we call edges uni-colored or bi-colored. We call a maximal connected chain of bi-colored triangles in G sharing bi-colored edges a

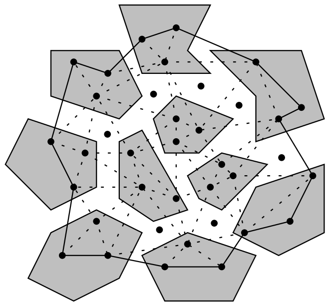

Figure 1.17: The sets S_i and the convex hull $\text{conv}_C(S)$ with respect to cone C. The $\mathcal{D}$-Delaunay triangulation is drawn dotted.

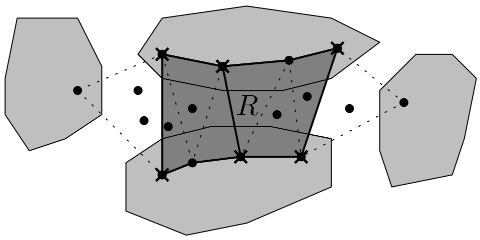

Figure 1.18: The corridor R which is split into two sub-corridors and two tri-colored triangles. The corners of the sub-corridors are marked by crosses.

corridor (cf. Figure 1.18). Since the graph G is planar and each of the induced subgraphs $G \cap D_i$ is connected according to Lemma 1.10 the number of such corridors is at most $3k - 6$. All colorless points are contained in the corridors and the tri-colored triangles because any uni-colored triangle is contained it its color-defining object. We break each corridor R into a minimum number of *sub-corridors*, i.e. sub-chains of the chain that forms R, so that the colorless points of each sub-corridor have measure at most $\delta \cdot \mu(S)$. Since the total number of corridors is $3k - 6$ the total number of sub-corridors is $O(1/\delta)$.

Each sub-corridor is bounded by two chains of uni-colored edges which we call *sides* and by two bi-colored edges which we call *ends* of the sub-corridor. The endpoints of the sides are called *corners* (cf. Figure 1.18). Let $N \subseteq S$ be the set of all corners of all sub-corridors. Since each sub-corridor has at most 4 corners the size of N is $O(1/\epsilon)$. The set N is an epsilon-net for the set of non-piercing objects $\mathcal{D}$.

The proof that N is indeed a weighted epsilon-net for the measure μ relies in principle on the fact that the collection $\mathcal{D}$ are non-piercing objects and follows along the lines of [MSW90].

Proof: Let D be an object that has no points of S on its boundary (cf. Lemma 1.11) and assume that D does not contain any points from N. The theorem is proven when we can show that the measure μ of D is less than

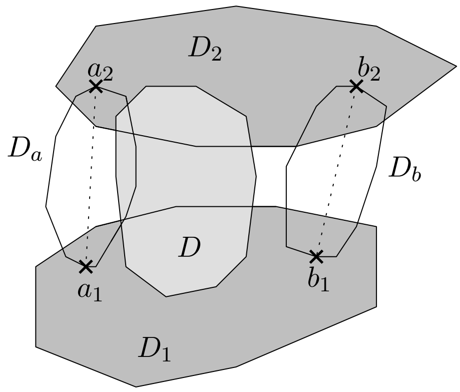

Figure 1.19: The case where D contains colored points of at least two colors.

$\epsilon \cdot \mu(S)$. If D contains no colored point then we are done, because the sets S_i were maximal. Hence, D must contain at least one colored point. If it contains two colored points, lets say z_1 of color 1 and z_2 of color 2, we can draw the following picture: Let D_1 be the color defining object of color 1 and D_2 the color defining object of color 2. Then D intersects D_1 and D_2 but cannot pierce them. The area between D_1 and D_2 is a sub-corridor whose ends we denote by (a_1, a_2) and (b_1, b_2). Lemma 1.12 assures that there is an object D_a that has a_1 and a_2 on its boundary and there is an object D_b that has b_1 and b_2 on its boundary. Since D also does not contain any point from N which are the corners of the sub-corridors, i.e. it does not contain a_1, a_2, b_1 or b_2 and since D and D_a as well as D and D_b are non-piercing it must lie between two ends of one sub-corridor. See Figure 1.19 for an illustration. Now, as all objects D_1, D_2, D_a and D_b have measure at most $\delta \cdot \mu(S)$ and the sub-corridor also has measure at most $\delta \cdot \mu(S)$, D can have measure of at most $5 \cdot \delta\mu(S) = 5/6\epsilon \cdot \mu(S) < \epsilon \cdot \mu(S)$.

The case where D only contains points of one color and colorless points is very similar. There is basically only one setup and it is depicted in Figure 1.20. Arguing as above it is easy to see in this case that D cannot have a measure of more than $4 \cdot \delta \cdot \mu(S) < \epsilon \cdot \mu(S)$.

□

Hence, we have the following theorem

Theorem 1.13 *Let $\mathcal{D}$ be the set of non-piercing objects in $\mathbb{R}^2$, that is produced by the projection τ. For every finite point set in non-$\mathcal{D}$-degenerate position there exists a weighted epsilon-net of size $O(1/\epsilon)$.*

Together with Lemma 1.3 and Lemma 1.5 this immediately implies our central theorem of this section.

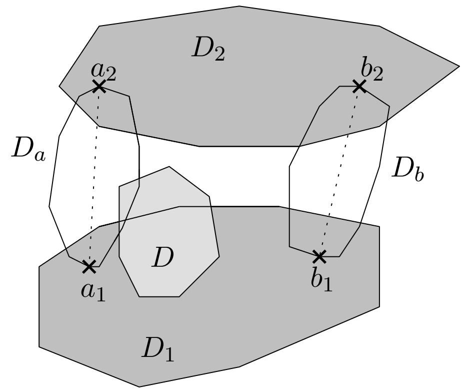

Figure 1.20: The case where D contains colored points of exactly one color.

Theorem 1.14 *Given a finite point set $P \subseteq \mathbb{R}^3$ and a polytope $T \subseteq \mathbb{R}^3$. The set system $(P, \mathcal{T})$ induced by a set of translates of polytope T admits an epsilon-net of size $O(1/\epsilon)$.*

1.2 From Epsilon-Nets to Hitting Sets

In this section we will describe a constant factor approximation algorithm to the hitting set problem using the epsilon-net of size $O(1/\epsilon)$ from the previous section. Recall that in the hitting set problem we are given a set of points $P \in \mathbb{R}^3$ and a set of polytopes that are all translates of the same polytope and we would like to select a subset $H \subseteq P$ of the input points of minimal cardinality such that every polytope is stabbed by a point in H. We denote the corresponding set system by $(P, \mathcal{T})$. The fractional hitting set problem is a relaxation of the original hitting set problem and is defined by the following linear program (LP1):

$$\begin{array}{lll} \min & \sum_{p\in P} x(p) & \\ \text{s.t.} & \forall T \in \mathcal{T} & \sum\limits_{p\in T} x(p) \geq 1 \\ & \forall p \in P & x(p) \geq 0 \end{array}$$

Let OPT denote the optimal size of the hitting set and OPT* the optimal value of the fractional hitting set problem. It is known that the integrality gap is constant for set systems that admit an epsilon-net of size $O(1/\epsilon)$ [PA95]. We will see a simple proof for this fact later in this section.

There are algorithms that compute a hitting set provided one has an algorithm that finds a small epsilon-net. The core idea to all these algorithms is to find a measure $\mu : P \to \mathbb{R}_{\geq 0}$ that assigns measures to the elements of P and finds an appropriate ϵ such that every set in $\mathcal{T}$ has a measure of at least $\epsilon \cdot \mu(S)$.

Once such a measure distribution is found it is then obvious that a weighted epsilon-net to this set system is automatically a hitting set.

The algorithm given by Brönnimann and Godrich [BG94] computes these measures iteratively. Initially, all elements have measure 1. Then, in each iteration an epsilon-net is computed and then checked whether it is also a proper hitting set. If not, i.e. there is a set which is not hit, then the measure of its elements is doubled. This is done until a hitting set is found. This algorithm can be seen as a deterministic analogue of the randomized natural selection technique used for instance by Clarkson [Cla95].

Another algorithm is by Even et al. [ERS05]. Here, the measure of the elements and ϵ are directly found by the following linear program (LP2):

$$\begin{aligned} & & & \max \epsilon \\ \text{s.t.} & \quad \forall T \in \mathcal{T} & & \mu(T) \geq \epsilon \\ & & & \sum_{p \in P} \mu(p) = 1 \\ & \quad \forall p \in P & & \mu(p) \geq 0 \end{aligned}$$

Let the optimal value of this linear program be ϵ^*.

It suffices to approximate the solution to this linear problem. There are numerous algorithms that find an approximate solution to such a covering linear program efficiently [You95, GK98].

One can reduce the problem of finding a weighted epsilon-net to the unweighted case. One just makes multiple copies of a point according to its assigned measure and it can be shown that the cardinality of this multiset can be bounded by $2n$ [CV05]. Hence, an $\frac{\epsilon}{2}$-net for this set system gives a hitting set for the original hitting set problem. The size of this hitting set is then $O(2/\epsilon^*)$.

As observed by Even et al. [ERS05] the linear program (LP1) for the fractional hitting set problem and the linear program (LP2) for finding an appropriate ϵ and a measure μ are in fact equivalent. One has just to substitute ϵ by $\sum_{p\in P} x(p)$ and $\mu(p)$ by $x(p)\cdot\epsilon$. Hence, we have $\mathrm{OPT} = 1/\epsilon^*$ and the constructed epsilon-net which is also a hitting set has size $O(\mathrm{OPT}^*)$, i.e. we have found an integral solution whose size is a constant away from the fractional solution. Or in other words, the integrality gap is constant.

Finally, we can conclude this section with the following theorem.

Theorem 1.15 *There exists a polynomial time algorithm that computes a constant-factor approximation to the hitting set problem for translates of polytopes in* $\mathbb{R}^3$.

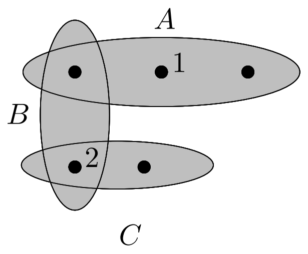

Figure 1.21: A set system with the subsets A, B and C.

1.3 From Hitting Set to Set Cover

The hitting set problem and the set cover problem are related to each other. In fact, they can be seen as dual to each other. The following definition states this relation precisely.

Definition 1.16 *The* dual set system *of a set system* $(P, \mathcal{T})$ *is the set system* $(\mathcal{T}, P^*)$ *where* $P^* = \{\mathcal{T}_p : p \in P\}$ *and* $\mathcal{T}_p$ *consists of all subsets of* $\mathcal{T}$ *that contain* p.

As an illustration Figure 1.21 depicts a set system with five points and the three subsets A, B, C. For the dual set system each of the subsets becomes a point and each point of the primal set system determines a subset in the dual. Hence, the corresponding dual set system is then formed by the points A, B, C and by the subsets $\{A\}$, $\{A, B\}$, $\{C\}$ and $\{B, C\}$. Notice that the dual of the dual is again the original set system.

Obviously, a hitting set for the primal set system is a set cover for the dual set system. Figure 1.21 for example illustrates this relation. The points labeled 1 and 2 form a hitting set. The corresponding subsets in the dual set system are $\{A\}$ and $\{B, C\}$. They in turn form a set cover for the dual set system. Hence, in order to solve the set cover problem for a set system it suffices to solve the hitting set problem for the dual set system. For arbitrary set systems, the dual set system can be of quite different structure. In general it is only known that the VC-dimension of the dual set system is less than 2^{d+1}, where d is the VC-dimension of the primal set system [Ass83].

However, we observe that if the set system is induced by translates of a polytope, then the dual is again induced by translates of a polytope. To see this, let $(P, \mathcal{T})$ be the primal set system. One just reduces each polytope $T \in \mathcal{T}$ to a point, for instance each to its lowest vertex. Let this be the set P'. Then, replace each point of P by a translate of the polytope T' which is the inversion of T in a point. One easily verifies that the so constructed set system $(P', \mathcal{T}')$ of points P' and collection of translates of polytope T' is indeed equivalent to the dual $(\mathcal{T}, P^*)$. This holds in fact for all $\mathbb{R}^d$. Figure 1.22 shows an example for planar polytopes.

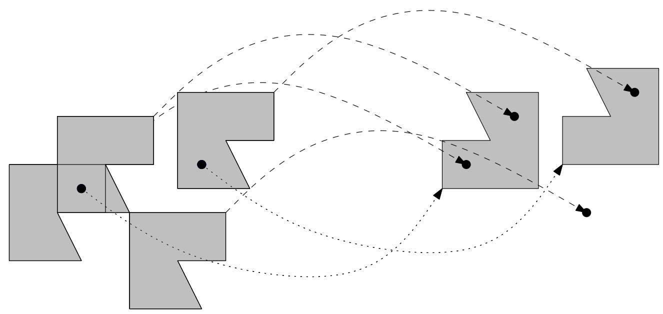

Figure 1.22: A set of points and planar polytopes is depicted on the left with the corresponding dual on the right. The dotted arrows show the correspondence between the points in the primal and the polytopes in the dual and the dashed arrows show the correspondence with the polytopes in the primal and the points in the dual.

Hence, we can find a constant-factor approximation to the set cover problem for translates of a polytope in $\mathbb{R}^3$ in polynomial time.

This brings us to our central theorem of this chapter.

Theorem 1.17 *There exists a polynomial time algorithm that computes a constant-factor approximation to the set cover problem for translates of polytopes in* $\mathbb{R}^3$.

1.4 Conclusions and Open Problems

In this chapter we have given the first constant-factor approximation algorithm for finding a set cover for a set of points in $\mathbb{R}^3$ by a given collection of translates of a polytope as well as the first constant-factor approximation algorithm for the corresponding hitting set problem. We achieved this result by providing an epsilon-net of size $O(\frac{1}{\epsilon})$ for the corresponding set system which is optimal up to a multiplicative constant. The scope of this chapter was to show that a constant factor approximation algorithm does exist. We did not focus on minimizing this constant, however.

Even though we can approximate a unit ball in $\mathbb{R}^3$ up to any given precision by a polytope, the corresponding question, whether there exists a constant-factor approximation algorithm for unit balls in $\mathbb{R}^3$ still remains open.

Chapter 2

Power Assignment Problems in Wireless Networks

2.1 Introduction

Wireless network technology has gained tremendous importance in recent years. The availability of high-bandwidth connections not only opens new application areas for mobile devices, but also replaces more and more so far 'wired' network installations. While the spatial aspect was already of interest in the wired network world due to cable costs etc., it has far more influence on the design and operation of wireless networks. The power required to transmit information via radio waves is heavily correlated with the Euclidean distance of sender and receivers. Hence problems in this area are prime candidates for the use of techniques from computational geometry.

In contrast to wired or cellular networks, ad hoc wireless networks a priori are unstructured in a sense that they lack a predetermined interconnectivity. An ad hoc wireless network is built of a set of radio stations P, each of which consists of a receiver as well as a transmission unit. A radio station p can send a message by setting its *transmission range* $r(p)$ and then by starting the transmission process. All other radio stations at distance at most $r(p)$ from p will be able to receive p's message (we are ignoring interference for now). For transmitting a message across a transmission range $r(p)$, the power consumption of p's transmission unit is proportional to $r(p)^\alpha$, where α is the *transmission power gradient*. In the idealistic setting of empty space, $\alpha = 2$, but it may vary from 2 to more than 6 depending on the environment conditions of the location of the network [Rap96, Pat00]. Given some transmission range assignment $r : P \mapsto \mathbb{R}_{\geq 0}$ for all nodes in the network, we can derive the so-called *communication graph* $G^{(r)} := G(P, E)$. $G(P, E)$ is a directed graph with vertex set P which has a directed edge (p, q) if and only if $r(p) \geq \|pq\|$, where

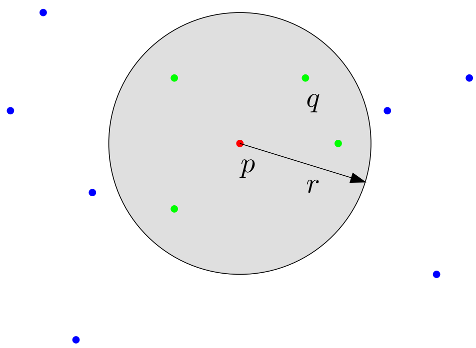

Figure 2.1: One transmission operation from node p to its neighboring nodes.

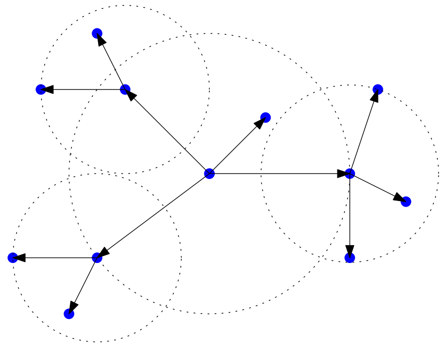

Figure 2.2: A power range assignment and the induced communication graph.

$\|pq\|$ denotes the Euclidean distance between p and q. Figures 2.1 and 2.2 illustrate the definition by an example. The energy consumption or cost of the transmission range assignment r is

$$\text{cost}(r) := \sum_{p \in P} r(p)^\alpha.$$

Since the sites often have limited power supply, the energy consumption of the communication is an important optimization criterion.

A fundamental problem in radio networks is that of assigning suitable powers to the individual network nodes such that (1) the resulting communication graph satisfies a certain connectivity property Π, and (2) the overall energy assigned to all the network nodes is minimized. Many properties Π can be considered and have been treated in the literature before; general surveys of algorithmic range assignment problems can be found in [CHP$^+$02, WNE00, KKKP00].

In this chapter we consider several definitions of Π to solve the following problems.

k-hop Broadcast: Given a set of stations and a specific source station s, we want the communication graph to contain a directed spanning tree rooted at s of depth at most k.
k-Set Broadcast: Given a specific source node s we want to find a transmission range assignment r of minimum total cost such that the respective communication graph $G^{(r)}$ contains a directed spanning tree rooted at s and at most k nodes have a non-zero transmission range assigned.
k-hop Multicast: Given a set P of stations, a specific source station s, a set of clients/receivers $C \subseteq P$, and some constant k, we want the communication graph to contain a directed tree rooted at s spanning all nodes in C with depth at most k and possibly using some intermediate nodes from P.
k-Station Network/k-disk Cover: Given a set P of stations and some constant k, we want to assign transmission powers to at most k stations (senders)

such that every station in P can receive a signal from at least one sender. We distinguish two cases: In the *non-discrete case* we are allowed to place the k senders anywhere in the space and in the *discrete case* the senders must be from the input set P.
TSP under squared Euclidean distance: Given a set P of n stations, determine a permutation $p_0, p_1, \dots p_{n-1}$ of the nodes such that the total energy cost of the TSP tour, i.e. $\sum_{i=0}^{n-1} \|p_i p_{(i+1) \bmod n}\|^\alpha$ is minimized.

Since all the above problems are NP-hard or believed to be NP-hard we look for efficient approximation schemes to solve them. The efficiency of our algorithms that we will present relies on the fact that for many of the above mentioned problems we are able to derive small *coresets*, a powerful concept from computational geometry. That is, for a given instance we identify a small subset of the original problem instance for which the solution translates to an almost as good solution of the original problem. By this, we achieve a speedup in running time compared to previous algorithms by orders of magnitude.

For analytical purposes it is very convenient to assume that all network nodes are placed in the Euclidean plane. Unfortunately, for real-world wireless network deployments, especially if not in the open field, the experienced energy requirement to transmit does not exactly correspond to some power of the Euclidean distance between the respective nodes. Buildings, uneven terrain or interference might affect the transmission characteristics considerably. Nevertheless there is typically still a strong correlation between the actual geographic distance and the required transmission power. An interesting question is now how to model analytically this correlation between geographic distance and energy requirement. We will look at this question in the next chapter more closely. For now, we assume that the radio stations are located in the Euclidean plane and that the energy for a transmission operation corresponds to some power of the Euclidean distance. In the next chapter we will see how to adapt our algorithms to a more general setup.

2.2 Bounded-hop Energy-efficient Broadcast in $\mathbb{R}^2$

One of the most basic communication tasks in wireless radio networks is *broadcasting*. That is, given a set of radio nodes P and a specific source node $s \in P$ we want to broadcast a message originating from s to all other radio nodes in the network.

There are two simple approaches to this. First, the radio station s sends with such a strong signal that it reaches all other radio stations in the network, see Figure 2.3 for an example. The advantage of this approach is that all radio stations receive the signal without any delay. However, the big disadvantage is the energy consumption. In order to submit the signal to all other radio stations, s has to set its transmission range $r(s)$ appropriately high. Since the energy consumption is proportional to $r(s)^\alpha$ where α is usually between 2 and

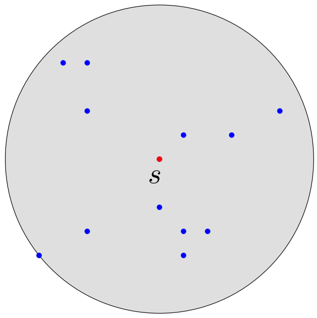

Figure 2.3: The source node s broadcasts its message to all other nodes within one hop.

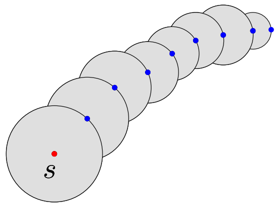

Figure 2.4: The source node s broadcasts its message to all other nodes using intermediate nodes.

6, the power that s has to send with is very high. Another approach for solving the broadcasting problem is that s sends to its immediate neighbors and they in turn send to their neighbors and so on (cf. Figure 2.4). The clear advantage of this approach is the energy efficiency. It uses much less energy than the first method. On the other hand, the big drawback of this approach is the latency. We do not have any control over the time delay by which the last radio node has received the signal. In this case we are not able to give any guarantee for the latency.

By finding a solution in between the two described approaches we can combine the advantages of them and overcome the disadvantages of both approaches. That is, we use intermediate radio nodes to broadcast a message from s but we restrict the number of intermediate radio nodes from s to any other radio node to a constant number k. We output a solution that obeys this hop restriction and is energy-minimal. In other words, given a particular source node s the communication graph $G^{(r)}$ must contain a directed spanning tree rooted at source s to all other nodes $p \in P$ having depth at most k.

Related Work

The general broadcast problem – assigning powers to stations such that the resulting communication graph contains a directed spanning tree and the total amount of energy used is minimized– has a long history. The problem is known to be NP-hard for $\alpha > 1$ ([CCP$^+$01, CHP$^+$02]), and for arbitrary, non-metric distance functions the problem can also not be approximated better than a log-factor unless P = NP [SK99]. For the Euclidean setting in the plane, the minimum spanning tree (MST) induces a transmission range assignment. A lower bound of 6 for the approximation ratio of the MST-based solution has been shown in [CCP$^+$01] and [WCLF01]. In a sequence of papers the upper bound for this solution was lowered in several steps (e.g. [CCP$^+$01, WCLF01, FNKP04]) to finally match its lower bound of 6 ([Amb05]). There is no other polynomial time algorithm known with a better approximation guarantee. While all these papers focused on analytical worst-case bounds for the algorithm performance, simulation studies e.g. in [CHR$^+$03] show that the actual performance in "real-world" networks is much better.

There has also been work done on the bounded hop broadcast problem. In [ACI$^+$04] Ambühl et al. present an exact algorithm for solving the 2-hop broadcast problem with a running time of $O(n^7)$ as well as a polynomial-time approximation scheme for a fixed number k of hops and constant ϵ which has running time $O(n^\mu)$ where $\mu = O((k^2/\epsilon)^{2^k})$, that is, their algorithm is *triply* exponential in the number of hops (and this dependence shows up in the exponent of n).

Our Contribution

This section contains two main contributions: We provide a coreset construction for the bounded-hop broadcast problem of size $O\left(\left(\frac{1}{\epsilon}\right)^{4k}\right)$; using this we obtain a $(1+\epsilon)$-approximation algorithm for energy-minimal broadcast whose running time is $O\left(\left(\frac{4k}{\epsilon}\right)^{\left(\frac{1}{\epsilon}\right)^{4k}} + n\right)$ in the Euclidean plane, that is, it is linear in n and the dependence on k is only doubly exponential.

2.2.1 Coresets

Without loss of generality we assume the largest Euclidean distance between the source node s and any other node $p \in P$ to be equal to 1. We say a range assignment r is *valid* if the induced communication graph $G^{(r)}$ contains a directed spanning tree rooted at s with depth at most k; otherwise we call r *invalid*.

Definition 2.1 (Coreset) *Let P be a set of n points, $s \in P$ a designated source node. Consider another set S of points (not necessarily a subset of P). If for any valid range assignment $r : P \mapsto \mathbb{R}_{\geq 0}$ there exist a valid range assignment $r' : S \mapsto \mathbb{R}_{\geq 0}$ such that $\text{cost}(r') \leq (1+\epsilon) \cdot \text{cost}(r)$ and for any valid range assignment $r' : S \mapsto \mathbb{R}_{\geq 0}$ there exists a valid range assignment $r : P \mapsto \mathbb{R}_{\geq 0}$ such that $\text{cost}(r) \leq (1+\epsilon) \cdot \text{cost}(r')$ then S is called (k,ϵ)-**coreset** for (P, s).*

A (k,ϵ)-coreset for a problem instance (P, s) can hence be viewed as a problem sketch of the original problem. If we can show that a coreset of small size (independent of n) exists, solving the bounded-hop broadcast problem on this problem sketch immediately leads to an $(1+\epsilon)^2$ solution to the original problem. The former can even be done using an exhaustive search algorithm.

This definition of a coreset differs slightly from the definition of a coreset defined in previous papers. For example, the term coreset has been defined for k-median [HM04] or minimum enclosing disk [KMY03]. However, in the case of the bounded-hop broadcast problem we have to consider two more issues. The first is feasibility. While any solution to the k-median problem is feasible not every solution is feasible for the bounded-hop broadcast problem. The second issue is monotonicity. For the problem of the smallest enclosing disk the optimal solution does not increase if we remove points from the input. We do not have this property here. An optimal solution can increase or decrease if we remove input points. Hence, the above definition of a coreset can be seen as the correct generalization to any optimization problem. We will make use of this definition throughout the whole chapter.

Our coreset construction is heavily based on the insight that for any valid range assignment r there exists an almost equivalent (in terms of total cost) range assignment r' where all assigned ranges are either zero or rather 'large'. We formalize this in the following structure lemma:

Lemma 2.2 (Structure Lemma) *Let r be a valid range assignment for (P, s) of cost $\text{cost}(r)$. For any $0 < \epsilon < 1$ there exists a valid range assignment r' with either $r'(p) = 0$ or $r'(p) \geq (1-\epsilon)\epsilon^{2k-2}$ and total cost $\text{cost}(r') \leq \left(1 + \frac{\epsilon}{1-\epsilon}\right)^{\alpha} \cdot \text{cost}(r)$.*

Proof: Let r be a valid range assignment. Consider a spanning tree rooted at s of depth at most k contained in the communication graph $G^{(r)}$. We call it the communication tree.

We will construct a valid range assignment r' from the given range assignment r. Initially, we set $r'(p) = r(p)$. After the first phase we will ensure $r'(s) \geq (1-\epsilon)\epsilon^{k-1}$ and after the second phase we will ensure $r'(p) \geq (1-\epsilon)\epsilon^{2k-2}$ for each node p.

The core idea to this construction is that if we have two nodes that are geometrically close to each other and one has a large power value $r(p)$ assigned to it and the other has a rather small power value, we can safely increase the larger a bit and remove the smaller one and still have a valid power assignment. We apply this idea once in the opposite direction of the communication paths, i.e. towards the source node s (first phase) and once along the direction of the directed communication paths (second phase).

If $r(s) \geq (1-\epsilon)\epsilon^{k-1}$ we are done with the first phase. Otherwise, there exists a directed path of length at least 1 from source node s to some node p having at most k hops. Let the nodes on this path be labeled $p = p_0, p_1, \ldots, p_l = s$, $l \leq k$ as in Figure 2.5.

Note that $r(p_0)$ does not contribute to the length of this path as it is the last node on the directed path. On this path pick the node with largest index j

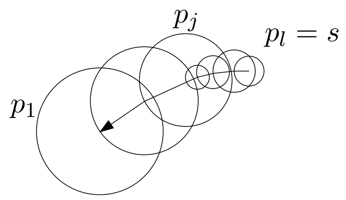

Figure 2.5: Original range assignment before the first phase

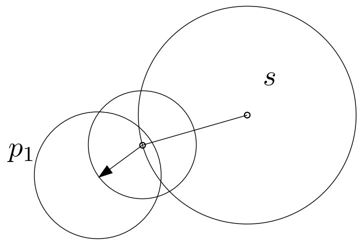

Figure 2.6: Range assignment after the first phase

such that $r(p_j) \geq (1-\epsilon)\epsilon^{j-1}$. Such a node clearly exists as $\sum_{i=1}^{l} r(p_i) \geq 1$ and $\sum_{i=1}^{l}(1-\epsilon)\epsilon^{i-1} < 1$. Setting $r'(s) = r(p_j)\left(1+\frac{\epsilon}{1-\epsilon}\right)$ and $r'(p_i) = 0$ for $i = j \ldots l-1$ increases the cost $\mathrm{cost}(r')$ only slightly but still ensures a valid range assignment because

$$\begin{aligned} r'(s) &= r(p_j)\left(1+\frac{\epsilon}{1-\epsilon}\right) \\ &\geq r(p_j) + \epsilon^j \\ &> r(p_j) + \sum_{i=j+1}^{l}(1-\epsilon)\epsilon^{i-1} \\ &> r(p_j) + \sum_{i=j+1}^{l} r(p_i), \end{aligned}$$

i.e. we have increased $r'(s)$ such that all nodes that could be reached by nodes $p_j, p_{j+1}, \ldots, p_{l-1}$ can now be reached directly by s.

In the second phase we can use an analogous argument starting from source node s. We assign each node p in the communication tree a level according to the number of hops to the source node s, where the source node s has level 0 and the leaves of the tree have level at most k.

We distinguish two cases. In the first case $r'(s) = r(s)$, i.e. the value of the starting node s has not been increased. The other case occurs when it has been increased, i.e. $r'(s) > r(s)$.

Let us look at the first case. Consider all maximal paths $\{t_j\}_j$ in the communication tree starting from node s where all nodes have $r(p) < (1-\epsilon)\epsilon^{k-1+i}$ if node p is on level i.

We can set $r'(s) = r(s)(1+\frac{\epsilon}{1-\epsilon})$ and $r'(p) = 0$ for all $p \in t_i$. Hence, we again maintain a valid range assignment and the next nodes p along the paths of the communication tree satisfy $r(p) \geq (1-\epsilon)\epsilon^{k-1+i}$ if node p is on level i. Applying the same reasoning iteratively to these nodes we finally have that for all nodes p either $r'(p) = 0$ or $r'(p) \geq (1-\epsilon)\epsilon^{k-1+i}$ for a node p on level i. Note that for nodes p on level k we can set $r'(p) = 0$. Hence, we have a valid range assignment

r' with $r'(p) \geq (1-\epsilon)\epsilon^{2k-2}$.

Let us now consider the second case, when $r'(s) > r(s)$, i.e. the value of s has been increased in the first phase of the construction. Here we increased $r'(s)$ already in the first phase to at least $(1-\epsilon)\epsilon^{k-2}\left(1+\frac{\epsilon}{1-\epsilon}\right) = \epsilon^{k-2}$. Hence, we can continue as in the first case without increasing $r'(s)$ anymore, because $\epsilon^{k-2} > \sum_{i=0}^{k}(1-\epsilon)\epsilon^{k-1+i}$ for $\epsilon < 1$.

The cost of the valid range assignment r' satisfies

$$\begin{aligned}
\text{cost}(r') &= \sum_{p\in P} (r'(p))^{\alpha} \\
&\leq \sum_{p\in P} \left(r(p)\left(1+\frac{\epsilon}{1-\epsilon}\right)\right)^{\alpha} \\
&= \left(1+\frac{\epsilon}{1-\epsilon}\right)^{\alpha} \cdot \text{cost}(r).
\end{aligned}$$

□

Using the preceding lemma it is now easy to come up with a small coreset. Intuitively we use a grid of width roughly an ϵ-fraction of the minimum non-zero range assigned in r'.

Lemma 2.3 *For any k-hop broadcast instance there exists a $(k,(\alpha+2)\epsilon)$-coreset of size $O\left(\left(\frac{1}{\epsilon}\right)^{4k}\right)$.*

Proof: We place a grid of grid width $\Delta = \frac{1}{\sqrt{2}}\epsilon \cdot r_{min}$, where $r_{min} = (1-\epsilon)\epsilon^{2k-2}$ on the plane. Notice, that the grid has to cover an area of radius 1 around the source only because the furthest distance from node s to any other node is 1. Hence its size is in $O\left(\left(\frac{1}{\epsilon}\right)^{4k}\right)$ for small ϵ. Now assign each point in P to its closest grid point. Let S be the set of grid points that had at least one point from P snapped to it.

It remains to show that S is indeed a coreset. We can transform any given valid range assignment r for P into a valid range assignment r'' for S in two steps. First, we apply the preceding Structure Lemma 2.2 to r to get a valid range assignment r' for P with $r'(p) = 0$ or $r'(p) \geq r_{min}$. If we define the range assignment r'' for S as

$$r''(p') = \max_{p \text{ was snapped to } p'} r'(p) + \sqrt{2}\Delta$$

we get a valid range assignment for S since all points in P are within $\Delta/\sqrt{2}$ of their respective nearest grid point. We then have

$$\begin{aligned} r''(p') &= \max_{p \text{ was snapped to } p'} r'(p) + \sqrt{2}\Delta \\ &\leq \max_{p \text{ was snapped to } p'} r'(p) + \epsilon \cdot r_{min} \\ &\leq (1+\epsilon) \max_{p \text{ was snapped to } p'} r'(p). \end{aligned}$$

Hence, the cost of the range assignment r'' satisfies

$$\text{cost}((r'') \leq (1+\epsilon)\,\text{cost}((r') \leq (1+\epsilon)\left(1+\frac{\epsilon}{1-\epsilon}\right)^{\alpha} \text{cost}(r).$$

On the other hand, we can transform any valid range assignment r'' for S into a valid range assignment r for P again in two steps. First, we use the Structure Lemma 2.2 to construct a valid range assignment r' for S with $r'(p) = 0$ or $r'(p) \geq r_{min}$. Then, we select for each grid point $g \in S$ one representative g_P from P that was snapped to it. For the grid point to which s (the source) was snapped we select s as the representative. If we define the range assignment r for P as $r(g_P) = r'(g) + \sqrt{2}\Delta$ and $r(p) = 0$ if p does not belong to the chosen representatives, then r is a valid range assignment for P because every point is moved by the snapping by at most $\Delta/\sqrt{2}$. Hence, we have

$$r(g_P) = r'(p) + \sqrt{2}\Delta = r'(p) + \epsilon \cdot r_{min} \leq (1+\epsilon)r'(p).$$

The cost of the valid range assignment r then satisfies

$$\text{cost}(r) \leq (1+\epsilon)\,\text{cost}(r') \leq (1+\epsilon)\left(1+\frac{\epsilon}{1-\epsilon}\right)^{\alpha} \cdot \text{cost}(r'').$$

For small ϵ we have $(1+\epsilon)\left(1+\frac{\epsilon}{1-\epsilon}\right)^{\alpha} \leq (1+(\alpha+2)\epsilon)$. This shows that S is indeed a $(k,(\alpha+2)\epsilon)$ coreset for P. □

Unfortunately we are not aware of any efficient algorithm for computing even just a constant approximation to the bounded-hop broadcast problem. But since we were able to reduce the problem size to a constant independent of n, we can also employ an exhaustive search strategy to compute an optimal solution for the reduced problem (S,s), which in turn translates to an $(1+(\alpha+2)\epsilon)^2$-approximate solution to the original problem since the reduced problem (S,s) is a $(k,(\alpha+2)\epsilon)$-coreset. In fact, it is still much more efficient to solve the problem for the coreset with a simple exhaustive search algorithm than to solve the problem for the coreset with the algorithm by Ambühl et al. [ACI+04].

Let us now concentrate on solving the bounded-hop broadcast problem for the coreset S. When looking for an optimal, energy-minimal solution for S, it is obvious that each node needs to consider only $|S|$ different ranges. Hence, naively there are at most $|S|^{|S|}$ different range assignments to consider at all.

We enumerate all these assignments and for each of them we check whether the induced communication graph contains a directed spanning tree of depth at most k rooted at the grid point corresponding to the original root node s, that is whether the respective range assignment is valid; this can be done in time $|S|^2$. Of all the valid range assignments we return the one of minimal cost.

Assuming the floor function, i.e. we can compute the floor of a rational number in constant time, a $(k, (\alpha+2)\epsilon)$-coreset S for an instance of the k-hop broadcast problem for a set of n radio nodes in the plane can be constructed in linear time. Hence we obtain the following theorem:

Theorem 2.4 *A $(1 + (\alpha + 2)\epsilon)^2$-approximate solution for the k-hop energy-minimal broadcast problem on n points in the plane can be computed in time*

$$O(n + |S|^{|S|}) = O\left(n + \left(\frac{1}{\epsilon}\right)^{4k(\frac{1}{\epsilon})^{4k}}\right).$$

A simple observation allows us to improve the running time slightly. Since eventually we are only interested in an approximate solution to the problem, we are also happy with only approximating the optimum solution for the coreset S. Such an approximation for S can be found more efficiently by not considering all possible at most $|S|$ ranges for each grid point. Instead we consider as admissible ranges only 0 and $r_{\min} \cdot (1 + \epsilon)^i$ for $i \geq 0$. That is, the number of different ranges a node can attain is at most $1 + \log_{1+\epsilon} r_{min}^{-1} \leq \frac{4k}{\epsilon} \cdot \log \frac{1}{\epsilon}$ for $\epsilon \leq 1$. This comes at a cost of a $(1+\epsilon)$ factor by which each individual assigned range might exceed the optimum. The running time of the algorithm improves, though, which leads to our main result in this section:

Theorem 2.5 *A $(1 + (\alpha + 2)\epsilon)^3$-approximate solution for the k-hop energy-minimal broadcast problem on n points in the plane can be computed in time*

$$O\left(n + \left(\frac{4k}{\epsilon} \cdot \log \frac{1}{\epsilon}\right)^{|S|}\right) = O\left(n + \left(\frac{4k}{\epsilon}\right)^{(\frac{1}{\epsilon})^{4k}}\right).$$

A $(1+\psi)$-approximate solution can be easily obtained by choosing $\epsilon = \theta(\psi/\alpha)$.

A solution to the k-hop broadcast problem addresses the issue of energy-efficiency and latency. We can bound the delay until the last radio node has received the message and we know that the energy consumed by one broadcasting operation is minimal for this setup. However, we did not consider the issue of interference and reliability. We will do this in the next section.

2.3 Minimum-energy Broadcast with Few Senders

One problem that is particularly prominent for the MST–based solution is the fact that in the resulting transmission range assignment a very large fraction of the network nodes are transmitting (i.e. have non-zero transmission range). In the MST-based range assignment, at least $n/6$ nodes are actually senders during the broadcast operation (since the maximum degree of the minimum spanning tree of a set of points in the Euclidean plane is bounded by 6). This raises several critical issues: (a) The more network nodes are transmitting in the process of one broadcast operation, the more likely it is that some nodes in the network experience interference due to several nearby nodes transmitting at the same time (unless special precautions are taken that interference does not occur). (b) Every retransmission of a message implies a certain delay which is necessary to set up the transmission unit etc; that is, the more senders are involved in the broadcast operation, the higher the latency becomes. This effect is even amplified by the previous problem if due to interference messages have to be resent. (c) Network nodes are not 100% reliable; if for example the probability for a network node to operate properly is 99.9%, the probability for a network broadcast to fail, i.e. not all nodes receiving the message, is $1 - 0.999^{(n/6)}$, which is around 40% for a network of $n = 3000$ nodes. This suggests to look for broadcast operations in the network that use only very few sending nodes. Of course, this comes at the cost of an increased total power consumption, but the behavior with respect to the critical issues (a) to (c) can be drastically improved.

In this section we study the following restricted broadcast operation: Given a specific source node s we want to find a transmission range assignment r of minimum total cost such that the respective communication graph $G^{(r)}$ contains a directed spanning tree rooted at s and at most k nodes have a non-zero transmission range assigned. We call this problem the *k-set energy-minimal broadcast problem.* Allowing only a small number k of sending nodes during the broadcast operation has several advantages: (a) The k transmissions can be easily scheduled in k different time slots, hence avoiding any interference at all. (b) The latency is obviously bounded by $O(k)$. (c) In the above scenario the probability of a broadcast operation to fail is $1 - 0.999^k$, which e.g. for $k = 10$ is 1%.

In this section we consider the k-set minimum energy broadcast problem from an analytical point of view. We show that somewhat surprisingly again for any network of n radio stations there exists a subset S of the stations whose size is *independent* of n and which preserves all the important characteristics of P with respect to an energy efficient k-set broadcast. We call S a coreset of the network topology with respect to the k-set broadcast problem. In contrast to the k-hop broadcast problem where the coreset has a size exponential in k, we will derive a coreset for the k-set broadcast problem whose size is polynomial in k and $1/\epsilon$. In fact, we will show that using a coreset of size $|S| = O((k/\epsilon)^2)$, any solution of the k-set broadcast problem for S translates to a solution for the k-set broadcast problem for the original set P at a cost at most a $(1+\epsilon)$ factor away and vice versa. Since the size of this coreset is independent of the network size, we can even afford to run an exhaustive search algorithm to compute an optimal k-set broadcast. The running time of this algorithm is linear in n but still exponential in k. So we also present an $O(1)$-approximation algorithm whose running time is linear in n but polynomial in k.

Closely related in particular to the $O(1)$-approximation algorithm that we will present is the work by Bilò et al. [BCKK05]. They consider the problem of covering a set of n points in the plane using at most k disks such that the sum of the areas of the disks is minimized. They provide a $(1+\epsilon)$-approximation to this problem in time $O(n^{\alpha^2/\epsilon^2})$. They do not address the problem of enforcing connectivity which is part of the k-set broadcast problem.

Section 2.3.1 recaps a known complexity result for the unconstrained broadcast problem and sketches a simple folklore-brute-force algorithm to solve the k-set broadcast problem. Section 2.3.3 contains the core contributions for the k-set broadcast problem; we show how to extract a small coreset of the network topology and how to use that to obtain a $(1+\epsilon)$-approximation algorithm. In Section 2.3.4 we show how a faster algorithm obtains an $O(1)$-approximation.

2.3.1 Preliminaries

As mentioned before the unconstrained broadcast problem is known to be NP-hard for $\alpha > 1$ and for non-metric distance functions even not well approximable ([CCP$^+$01, SK99]. Since the unconstrained broadcast problem is a special case of the k-set broadcast problem with $k = n$ these hardness results carry over to the k-set broadcast problem, if k is not treated as a constant. If k is regarded a constant, the problem can be solved in polynomial time as we will see in the following.

2.3.2 A Naive Algorithm

The k-set broadcast problem can be solved in a naive way. Essentially, one can try out all $\binom{n}{k-1}$ different subsets for the $k-1$ active senders apart from the source s. For each of those (and the source node s), one then assigns all possible $n-1$ ranges. In total we have then $O(n^{k-1}(n-1)^k) = O(n^{2k})$ potential power assignments. For each of those we can check in $O(n^2)$ time whether it is a valid k-set broadcast.

That is, overall we have the following corollary:

Corollary 2.6 *For n points we can compute the optimal k-set broadcast in time $O(n^{2k+2})$.*

For most practical applications, we expect k to be a small constant, but unfortunately not small enough that this naive algorithm can be applied to networks of not too small size (e.g. several thousand nodes). In the following we lower our expectations and aim for *approximate* solutions to the k-set broadcast problem. This allows for more efficient algorithms as we will see.

2.3.3 Small Coresets of the Network Topology

We will now show that we can find a small coreset to the original problem. We assume that the maximum distance from the source node s to another node is 1.

First, we need to show a technical lemma:

Lemma 2.7 *The term*

$$\frac{\sum_{i=1}^{k}(r_i+\delta)^\alpha}{\sum_{i=1}^{k} r_i^\alpha}$$

is maximized if $r_1 = r_i$ *for all* $i \in \{2, \ldots, k\}$ *for* $r_i, \delta \geq 0$

Proof: Let us first consider the case for two variables: We claim that the expression

$$\max_{r_1^\alpha + r_2^\alpha = c} (r_1+\delta)^\alpha + (r_2+\delta)^\alpha$$

attains its maximum when $r_1 = r_2$. Since $r_1^\alpha + r_2^\alpha = c$ we have $r_1 = (c - r_2^\alpha)^{1/\alpha}$. Thus we want to find the maximum of the function

$$f(r_2) := ((c - r_2^\alpha)^{1/\alpha} + \delta)^\alpha + (r_2 + \delta)^\alpha$$

We have

$$f'(r_2) = \alpha \cdot \left((r_2+\delta)^{\alpha-1} - \left(\frac{r_2 \cdot \left((c - r_2^\alpha)^{1/\alpha} + \delta \right)}{(c - r^\alpha)^{1/\alpha}} \right)^{\alpha-1} \right)$$

Thus

$$\begin{aligned} f'(r_2) = 0 \quad \Leftrightarrow \quad r_2 + \delta \quad &= \quad \frac{r_2 \cdot \left((c - r_2^\alpha)^{1/\alpha} + \delta \right)}{(c - r^\alpha)^{1/\alpha}} \\ \Leftrightarrow \quad r_2 + \delta \quad &= \quad r_2 \cdot \frac{r_2 \cdot \delta}{(c - r^\alpha)^{1/\alpha}} \quad \Leftrightarrow \quad r_2 \quad = \quad (c/2)^{1/\alpha} \end{aligned}$$

Since $r_1 = (c - r_2^\alpha)^{1/\alpha} = (c/2)^{1/\alpha}$ we have $r_1 = r_2$. Furthermore note that this is the only maximum of the function $f(r_2)$ inside its domain, and that the value of the function at the boundary is strictly less.

Now let us consider the case with more that two variables. We can restate the claim as follows: The expression

$$\max_{\sum_{i=1}^{k} r_i^\alpha = c} \sum_{i=1}^{k} (r_i + \delta)^\alpha$$

attains its maximum when $r_1 = r_i$ for all i. Assume otherwise, i.e. the expression is maximal when at least two variables have different value. Let these two variables be without loss of generality $r_1 \neq r_2$. However, we have

$$\max_{\sum_{i=1}^{k} r_i^\alpha = c} \sum_{i=1}^{k} (r_i + \delta)^\alpha = \max_{r_1^\alpha + r_2^\alpha + \sum_{i=3}^{k} r_i^\alpha = c} (r_1 + \delta)^\alpha + (r_2 + \delta)^\alpha + \sum_{i=3}^{k} (r_i + \delta)^\alpha.$$

However, if we fix r_i for $3 \leq i \leq k$ and the sum $r_1^\alpha + r_2^\alpha$, we see from the two-variable case that the expression can be increased if $r_1 \neq r_2$. This contradicts the assumption that the maximum is attained when at least two variables have different value. Hence, for $r, \delta \geq 0$ the expression

$$\frac{\sum_{i=1}^{k} (r_i + \delta)^\alpha}{\sum_{i=1}^{k} r_i^\alpha}$$

is maximized if $r_1 = r_i$ for all $i \in \{2, \ldots, k\}$. □

Lemma 2.8 *For any k-set broadcast instance there exists a $(k, (1+\epsilon)^\alpha)$-coreset of size $O\left(\frac{k^2}{\epsilon^2}\right)$.*

Proof: We place a grid of grid width $\Delta = \frac{1}{\sqrt{2}} \frac{\epsilon}{k}$ on the plane. Notice, that the grid has to cover an area of radius 1 around the source only because the furthest distance from node s to any other node is 1. Hence its size is $O\left(\frac{k^2}{\epsilon^2}\right)$. Now we assign each point in P to its closest grid point. Let S be the set of grid points that had at least one point from P snapped to it.

It remains to show that S is indeed a $(k, (1+\epsilon)^\alpha)$-coreset. We can transform any given valid range assignment r for P into a valid range assignment r' for S. We define the range assignment r' for S as

$$r'(p') = \max_{p \text{ was snapped to } p'} r(p) + \sqrt{2}\Delta.$$

Since each point p is at most $\frac{1}{\sqrt{2}}\Delta$ away from its closest grid point p' we certainly have a valid range assignment for S. It is easy to see that the cost of r' for S is not much larger than the cost of r for P. We have:

$$\begin{aligned}
\sum_{p' \in S} (r'(p'))^\alpha &= \sum_{p' \in S} (\max_{p \text{ was snapped to } p'} r(p) + \sqrt{2}\Delta)^\alpha \\
&\leq \sum_{p' \in S} (\max_{p \text{ was snapped to } p'} r(p) + \frac{\epsilon}{k})^\alpha \\
&\leq \sum_{p \in P} (r(p) + \frac{\epsilon}{k})^\alpha.
\end{aligned}$$

The relative error satisfies

$$\frac{\mathrm{cost}(r')}{\mathrm{cost}(r)} \leq \frac{\sum_{p\in P}(r(p)+\frac{\epsilon}{k})^\alpha}{\sum_{p\in P}(r(p))^\alpha}.$$

Notice, that $\sum_{p\in P} r(p) \geq 1$ and r is positive for at most k points p. Hence, the above expression is maximized when all $r(p)$ have the same value r_p for all points p that are assigned a positive value, see Lemma 2.7. Thus

$$\frac{\mathrm{cost}(r')}{\mathrm{cost}(r)} \leq \frac{k\cdot(\frac{1}{r_p}+\frac{\epsilon}{r_p})^\alpha}{k\cdot(\frac{1}{r_p})^\alpha} = (1+\epsilon)^\alpha.$$

On the other hand we can transform any given valid range assignment r' for S into a valid range assignment r for P as follows. We select for each grid point $g \in S$ one representative g_P from P that was snapped to it. For the grid point to which s (the source) was snapped we select s as the representative. If we define the range assignment r for P as $r(g_P) = r'(g) + \sqrt{2}\Delta$ and $r(p) = 0$ if p does not belong to the chosen representatives, then r is a valid range assignment for P because every point is moved by the snapping by at most $\Delta/\sqrt{2}$. Using the same reasoning as above we can show that $\mathrm{cost}(r) \leq (1+\epsilon)^\alpha \mathrm{cost}(r')$. Hence, we have shown that S is indeed a $(k,(1+\epsilon)^\alpha)$-coreset. □

Once we have solved the k-set broadcast problem for the $(k,(1+\epsilon)^\alpha)$-coreset S we can easily transform the obtained solution to a $(1+\epsilon)^{2\alpha}$-approximate solution to the original problem. Let us now concentrate on solving the k-set broadcast problem for the coreset S. Since we were able to reduce the problem size to a constant independent of n, we can employ an exhaustive search strategy to compute an optimal solution for the reduced problem (S,s).

When looking for an energy-minimal solution for S, it is obvious that we need to consider only $|S|$ different ranges for each node. Hence, naively there are at most $\binom{\frac{k^2}{\epsilon^2}}{k}\cdot\left(\frac{k^2}{\epsilon^2}\right)^k$ different range assignments to consider at all. We enumerate all these assignments and for each of them we check whether the range assignment is valid; this can be done in time $|S|^2$. Of all the valid range assignments we return the one of minimal cost.

Hence, a $(k,(1+\epsilon)^\alpha)$-coreset S for an instance of the k-set broadcast problem for a set of n radio nodes in the plane can be constructed in linear time, provided we can compute the floor of a rational number in constant time. Hence we obtain the following theorem:

Theorem 2.9 *A $(1+\epsilon)^{2\alpha}$-approximate solution for the k-set broadcast problem on n points in the plane can be computed in time $O\left(n+\left(\frac{k}{\epsilon}\right)^{4k+4}\right)$.*

Again we can improve the running time slightly. Since eventually we are only interested in an approximate solution to the problem, we only need to approximate the optimum solution for the coreset S. Such an approximation for S can be found more efficiently by not considering all possible at most $|S|$ ranges for each grid point. Instead we consider as admissible ranges only 0 and $\frac{\epsilon}{k} \cdot (1+\epsilon)^i$ for $i \geq 0$. That is, the number of different ranges a node can attain is at most $1 + \log_{1+\epsilon} \frac{k}{\epsilon} \leq \frac{2}{\epsilon} \cdot \log \frac{k}{\epsilon}$ for $\epsilon \leq 1$. This comes at a cost of a $(1+\epsilon)$ factor by which each individual assigned range might exceed the optimum. The running time of the algorithm improves, though, which leads to our main result in this section:

Theorem 2.10 *A $(1+\epsilon)^{3\alpha}$-approximate solution for the k-set broadcast problem on n points in the plane can be computed in time $O\left(n + \frac{k^4}{\epsilon^4}\left(\frac{k^2 \log \frac{k}{\epsilon}}{\epsilon^3}\right)^k\right)$.*

2.3.4 Faster $O(1)$-Approximations

We now show how to compute a constant approximation for the k-set broadcast problem. The idea is to first cluster the points into k clusters. Then we ensure connectivity of these point sets by increasing their cluster sizes. As clustering we define the *k-disk cover problem*:

Definition 2.11 (k-disk cover problem) *Given a set P of n points in the Euclidean plane, find a subset $C \subseteq P$ of cardinality at most k and radii $r_p \geq 0$ associated with each element $p \in C$ such that $\sum_{p \in C} r_p^\alpha$ is minimized and all points in P are covered by the disks $D_p^{r_p} := \{x \in \mathbb{R}^2 \mid \|xp\| \leq r_p\}$.*

Given a k-disk cover $D := (C,\ (r_p)_{p \in C})$ for P with center points C and radii r_p, we associate with D a range assignment r_D on P as follows:

$$\forall p \in P: \quad r_D(p) := \begin{cases} r_p & \text{if } p \in C \\ 0 & \text{otherwise} \end{cases}$$

By D_i we denote a disk in D. Note that the k-disk cover problem with the additional constraint that the communication graph $G^{(r_D)}$ is connected is exactly the k-set broadcast problem and that an instance of one problem is a relaxation of the of the other. Thus, the cost of an optimal solution for an instance of the k-disk cover problem is a lower bound for the k-set broadcast problem. Unfortunately, the k-disk cover problem is NP-hard (see [BCKK05]) but it admits a PTAS as shown by Bilò et al. [BCKK05]. A direct consequence of their results is:

Corollary 2.12 *There exists an algorithm for the k-disk cover problem in $\mathbb{R}^d$ that computes $(1+\epsilon)$-approximate solution in time $n^{(\frac{\alpha}{\epsilon})^{O(d)}}$.*

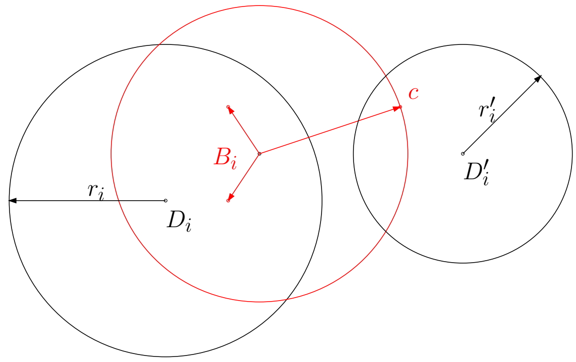

Figure 2.7: Proof illustration for the constant factor approximation algorithm.

By setting ϵ to 1 we obtain a 2-approximation algorithm for the k-disk cover problem that runs in time $n^{c'_\alpha}$ for a constant c'_α. A brief sketch of this algorithm can be found in the next section. Note that the algorithm can easily be modified such that the source s is the center of one of the disks.

Our approximation algorithm works as follows: First we compute an approximate k-disk cover $D := (C,\ (r_p)_{p \in C})$ over P. Then we determine for the center points in C an approximate broadcast with range assignment r_B by using an minimum spanning tree based algorithm (see [Amb05]) that has an approximation guarantee of 6. Now we construct a range assignment r_A for P in the following way:

$$\forall p \in P: \quad r_A(p) := \begin{cases} \max\{r_p,\ r_B(p)\} & \text{if } p \in C \\ 0 & \text{otherwise} \end{cases}$$

Note that $G^{(r_A)}$ is connected and therefore induces a valid k-set broadcast since only k stations are sending. We still have to show that we have computed an approximate solution:

Theorem 2.13 $\text{cost}(r_A) \leq 36 c_\alpha \cdot \text{cost}(r_{opt})$, *where* r_{opt} *is the range assignment of an optimal* k*-set broadcast and* c_α *is a constant depending only on* α*.*

Proof: The proof idea is the following: Assuming knowledge about an optimal range assignment r_{opt} for the k-set broadcast we transform the range assignment r_D into r'_D such that a) r'_D is a valid k-set broadcast b) the sending nodes in r'_D are exactly the center points of D and c) r'_D is a constant factor approximation of r_{opt}. If we know that such a broadcast r'_D exists, we can simply compute an optimal broadcast r_B over the center points of D. Then we know that $\text{cost}(r_B) \leq \text{cost}(r'_D)$ and r_B must also be a constant factor approximation of r_{opt}.

Consider now the communication tree T which is defined as a subtree of $G^{(r_{opt})}$ spanning P. The idea of the construction of r'_D is to replace the inner nodes of T (i.e. the sending stations of r_{opt}) by increasing the radii of the disks in C appropriately so that r'_D is valid.

We increase the nonzero values of r_D in the following way: with each of the inner nodes B_i of T we associate arbitrarily one disk D_i in which B_i is contained. Note that there must be at least one such disk for each B_i since the disks cover all input points P. We now update r_D in a breath first search manner on T starting from source node s (see Figure 2.7):

Given an inner node B_i of T if all children of B_i in T lie in the associated disk D_i then all of them can be reached from node C_i without increasing r_i. The interesting case is if there are children of B_i that are not contained in D_i but contained in a disk D'_i whose center is not covered so far. Assume that there is exactly one such child c. We then set the radius of D_i to $r_i + r'_i + r_{opt}(B_i)$. If there is more than one such child, let c be the one that maximizes r'_i so that each child of B_i and the centers of the disks in which the children of B_i are contained in can be reached by D_i. Note that it can happen that two different inner nodes B_i and B_j are associated with the same disk D_k, so that D_k is updated more than once in the process. In such a case we update D_k only if r_k is increasing. Now let us assume that for a disk D_i the last update involved disk D'. We then call disk D' the target disk of D_i.

By induction $G^{(r_D)}$ is connected after these updates. Furthermore, note that the sending stations are still exactly the center points of D. Let $D^* \subseteq D$ be the set of disks that are updated and let B_i be the node in T in the update step for disk $D_i \in D^*$. Summing over all disks, the total cost of the broadcast is therefore bounded by:

$$\underbrace{\sum_{D_i \in D \setminus D^*} r_i^\alpha}_{\leq \text{cost}(D) \leq 2\,\text{cost}(r_{opt})} + \underbrace{\sum_{D_i \in D^*} (r_i + r'_i + r_{opt}(B_i))^\alpha}_{(**)}$$

Before we bound the second term, note that a disk appears as a target disk only once in the process of updating the disk radii since once its center point is covered it is never considered as a target disk again. Thus each r'_i in the above sum can also appear only once. Hence,

$$\begin{aligned}
(**) &\leq c_\alpha \left(\sum_{D_i \in D^*} r_i^\alpha + \sum_{D_i \in D^*} {r'_i}^\alpha + \sum_{D_i \in D^*} r_{opt}(B_i)^\alpha \right) \\
&\leq c_\alpha (\underbrace{2 \sum_{D_i \in D} r_i^\alpha}_{\leq 4\,\text{cost}(r_{opt})} + \underbrace{\sum_{D_i \in D} r_{opt}(B_i)^\alpha}_{=\text{cost}(r_{opt})}) \\
&\leq 5 \cdot c_\alpha \cdot \text{cost}(r_{opt})
\end{aligned}$$

where the constant c_α can be bounded by 3^α. Thus, there exists a broadcast on the center points C with a total cost that can be upper bounded by

$$\begin{aligned} & 2\,\mathrm{cost}(r_{opt}) + 5 \cdot c_\alpha \cdot \mathrm{cost}(r_{opt}) \\ \leq \; & 6 \cdot c_\alpha \cdot \mathrm{cost}(r_{opt}) \end{aligned}$$

Since we use a 6-approximate broadcast, the algorithm has an approximation ratio of $36c_\alpha$. □

Theorem 2.14 *There exists a constant factor approximation algorithm for the k-set broadcast problem on n points in the Euclidean plane that runs in $O(n^{c'_\alpha})$.*

The theorem can be further improved by using the results of the previous section. By setting ϵ to 1 we obtain a coreset of size k^2. Using Theorem 2.14 we obtain directly a constant factor approximation algorithm whose running time is only linear in n and polynomial in k:

Theorem 2.15 *There exists a constant factor approximation algorithm for the k-set broadcast problem on n points in the Euclidean plane that runs in time linear in n and polynomial in k, i.e. in $O(n + k^{c'_\alpha})$.*

2.4 Bounded-hop Multicast or: "Reaching Few Receivers Quickly"

Given a set P of n points (stations) in $\mathbb{R}^2$, a distinguished source point $s \in P$ (sender), and a set $C \subseteq S$ of client points (receivers) we want to assign ranges $r : S \mapsto \mathbb{R}_{\geq 0}$ to the elements in P such that the resulting communication graph contains a tree rooted at s spanning all elements in C and with depth at most k. The goal is to minimize the total assigned energy $\sum_{p \in P} r(p)^\alpha$. This can be thought of as the problem of determining an energy efficient way to quickly (i.e. within few transmissions) disseminate a message or a data stream to a set of few receivers in a wireless network.

As in the previous section we will solve this problem by first deriving a coreset S of size independent of $|P| = n$ and then invoking an exhaustive search algorithm. We assume both k and $|C| = c$ to be constants. The resulting coreset will have size *polynomial* in $1/\epsilon$, c and k. For few receivers this is a considerable improvement over the exponential-sized coreset that was used in Section 2.2 for the k-hop broadcast problem.

2.4.1 Algorithms

We will use the same coreset construction as for the k-set broadcast problem. If we would like to reach c receivers each within at most k hops we immediately know that at most kc senders can be actively sending. Hence, we can use the coreset construction for the k-set broadcast problem with kc active senders. Obviously, the so constructed coreset with respect to the k-set broadcast problem is also a coreset with respect to the k-hop multicast problem and has size $O\left(\frac{(kc)^2}{\epsilon^2}\right)$. All we need to do is to solve the problem for the coreset.

As we are not aware of any algorithm to solve the k-hop multicast problem we employ a naive exhaustive search strategy, which we can afford since after the coreset computation we are left with a constant problem size. Essentially we consider every kc-subset of S as potential set of senders and try out the $|S|$ potential ranges for each of the senders. Hence, naively there are at most $\binom{\frac{kc^2}{\epsilon^2}}{kc} \cdot \left(\frac{kc^2}{\epsilon^2}\right)^{kc}$ different range assignments to consider at all. We enumerate all these assignments and for each of them we check whether the range assignment is valid with respect to the set C' of grid points that have at least one point from C snapped to it; this can be done in time $|S|^2$. Of all the valid range assignments we return the one of minimal cost.

A coreset S for an instance of the k-hop multicast problem can be constructed in linear time. Hence we obtain the following theorem:

Theorem 2.16 *A $(1+\epsilon)$-approximate solution for the k-hop multicast problem on n points in the plane can be computed in time $O\left(n + \left(\frac{kc}{\epsilon}\right)^{4kc+4}\right)$.*

As we are only interested in an approximate solution, we do not have to consider all $|S|$ potential ranges but can restrict to essentially $O\left(\log_{1+\epsilon}\frac{kc}{\epsilon}\right)$ many. The running time of the algorithm improves accordingly:

Theorem 2.17 *A $(1+\epsilon)$-approximate solution for the k-hop multicast problem on n points in the plane can be computed in time $O\left(n + \frac{(kc)^4}{\epsilon^4}\left(\frac{(kc)^2\log\frac{kc}{\epsilon}}{\epsilon^3}\right)^{kc}\right)$.*

2.5 Energy-minimal Network Coverage or: "How to Cover Points by Disks"

The problem that we consider in this section was studied before in [AAB$^+$06, BCKK05] and aims to select and assign powers to k out of a total of n wireless network stations such that all stations are within reach of at least one of the selected stations and the required energy is minimal, i.e. given a set P of points in $\mathbb{R}^d$ and some constant k, we want to find at most k d-dimensional balls with radii r_i that cover all points in S while minimizing the objective function $\sum_{i=1}^{k} r_i^\alpha$ for some power gradient $\alpha \geq 1$. The problem was shown to be NP-hard for $\alpha > 1$ [BCKK05] and solvable in polynomial time for $\alpha = 1$ [GKK$^+$08].

We again show that a coreset of size polynomial in k and $1/\epsilon$ exists. This enables us to improve the running time of the $(1+\epsilon)$-approximation algorithm by Bilo et al. [BCKK05] from $n^{((\alpha/\epsilon)^{O(d)})}$ to a running time that is linear in n. We also present a variant that is able to tolerate few outliers and runs in polynomial time for constant values of k and the number of outliers.

We distinguish two cases: the discrete case in which the ball centers have to be in P and the non-discrete case where the centers can be located arbitrarily in $\mathbb{R}^d$.

2.5.1 A Small Coreset for k-disk Cover

In this section we describe how to find a coreset of size $O\left(\frac{k^{2d/\alpha+1}}{\epsilon^d}\right)$, i.e. of size independent of n and polynomial in k and in $1/\epsilon$.

For now let us assume that we are given the cost of a λ-approximate solution P^λ for the point set P. We start by putting a regular d-dimensional grid on P with grid cell width Δ depending on P^λ. For each cell C in the grid we choose an arbitrary representative point in $P \cap C$. We denote by S the set of these representatives. We say that C is *active* if $P \cap C \neq \emptyset$. Note that the distance between any point in $S \cap C$ and the representative point of C is at most $\sqrt{d} \cdot \Delta$. In the following we write R^{OPT} for an optimal solution for any point set $R \subseteq P$. We obtain a solution S_P^{OPT} by increasing the disks in an optimal solution S^{OPT} by an additive term $\sqrt{d} \cdot \Delta$. Since each point in P has a representative in S with distance at most $\sqrt{d} \cdot \Delta$, S_P^{OPT} covers P. In the following we will show that (i) the cost of S_P^{OPT} is close to the cost of an optimal solution P^{OPT} for the original input set P and (ii) the size of the coreset S is small.

Theorem 2.18 *We have in the*

$$\begin{aligned} \textit{non-discrete case:} \qquad & \text{cost}(S_P^{OPT}) \leq (1+\epsilon)^\alpha \cdot \text{cost}(P^{OPT}) \\ \textit{discrete case:} \qquad & \text{cost}(S_P^{OPT}) \leq (1+\epsilon)^{2\alpha^2} \cdot \text{cost}(P^{OPT}) \end{aligned}$$

$$\textit{for} \quad \Delta := \frac{1}{\sqrt{d}} \cdot \frac{\epsilon}{k^{1/\alpha}} \left(\frac{\text{cost}(P^\lambda)}{\lambda}\right)^{1/\alpha}$$

Proof: Suppose S^{OPT} is given by k balls $(C_i)_{i\in\{1,\dots,k\}}$ with radii $(r_i)_{i\in\{1,\dots,k\}}$. Then

$$\frac{\text{cost}(S_P^{OPT})}{\text{cost}(P^{OPT})} = \frac{\sum_{i=1}^k (r_i + \sqrt{d} \cdot \Delta)^\alpha}{\text{cost}(P^{OPT})} =: (I)$$

One can easily show that this term is maximized when $r_1 = r_i \; \forall \, i \in \{2,\dots,k\}$

(see Lemma 2.7). Thus we have

$$\begin{aligned}
\frac{\mathrm{cost}(S_P^{OPT})}{\mathrm{cost}(P^{OPT})} &\leq k \cdot \frac{\left(\frac{\mathrm{cost}(S^{OPT})^{1/\alpha}}{k^{1/\alpha}} + \frac{\epsilon}{k^{1/\alpha}}\left(\frac{\mathrm{cost}(P^{\lambda})}{\lambda}\right)^{1/\alpha}\right)^{\alpha}}{\mathrm{cost}(P^{OPT})} \\
&= \frac{\left(\mathrm{cost}(S^{OPT})^{1/\alpha} + \epsilon\left(\frac{\mathrm{cost}(P^{\lambda})}{\lambda}\right)^{1/\alpha}\right)^{\alpha}}{\mathrm{cost}(P^{OPT})} \\
&\leq \left(\frac{\mathrm{cost}(S^{OPT})^{1/\alpha}}{\mathrm{cost}(P^{OPT})^{1/\alpha}} + \epsilon\right)^{\alpha} \quad =: (II)
\end{aligned}$$

Now let us distinguish between the non-discrete and the discrete case. In the non-discrete case $(II) \leq (1+\epsilon)$ following from the monotonicity of the problem, i.e. for all subsets $R \subseteq P:\ \mathrm{cost}(R^{OPT}) \leq \mathrm{cost}(P^{OPT})$. This can easily be seen, as each feasible solution for P is also a feasible solution for R. In the discrete case $\mathrm{cost}(S^{OPT})$ can be bigger than $\mathrm{cost}(P^{OPT})$ - but not much as we will see: Given an optimal solution P^{OPT} we transform P^{OPT} into a feasible solution P_S^{OPT} for S by shifting the ball centers to their corresponding representative point in S. Since some points can be uncovered now we have to increase the ball radii by an additional $\sqrt{d} \cdot \Delta$. Following exactly the same analysis as above, we get in this case that $\mathrm{cost}(P_S^{OPT}) \leq (1+\epsilon)^{\alpha} \cdot \mathrm{cost}(P^{OPT})$. Hence $\mathrm{cost}(S_P^{OPT}) \leq ((1+\epsilon)^{\alpha} + \epsilon)^{\alpha} \leq (1+\epsilon)^{2\alpha^2} \cdot \mathrm{cost}(P^{OPT})$. □

Knowing that the coreset S is a good representation of the original input set P we will show that S is also small.

Theorem 2.19 *The size of the computed coreset S is bounded by*

$$O\left(\frac{k^{\frac{d}{\alpha}+1} \cdot \lambda^{d/\alpha}}{\epsilon^{d}}\right).$$

Proof: Observe that the size of S is exactly given by the number of cells that contain a point of P. The idea is now to use an optimal solution P^{OPT} to bound the number of active cells. We can do so because any feasible solution for P covers all points in P and thus a cell C can only be active if such a solution covers fully or partially C. Thus the number of active cells cannot be bigger than the volume of such a solution divided by the volume of a grid cell.

To ensure that also the partially covered cells are taken into account we increase the radii by an additional term $\sqrt{d} \cdot \Delta$. Consider P^{OPT} given by k balls

$(C_i)_{i\in\{1,\dots,k\}}$ with radii $(r_i)_{i\in\{1,\dots,k\}}$. Then

$$\begin{aligned}
|S| &\leq \sum_{i=1}^{k} \frac{2^d \cdot \left(r_i + \sqrt{d}\cdot\Delta\right)^d}{\Delta^d} \\
&= (2\sqrt{d})^d \cdot \sum_{i=1}^{k} \left(1 + \frac{r_i}{\Delta'}\right)^d \\
&\overset{(*)}{\leq} (2\sqrt{d})^d \cdot \sum_{i=1}^{k} \left(1 + \frac{(k\cdot\lambda)^{1/\alpha}}{\epsilon} \cdot \left(\frac{\mathrm{cost}(P^{OPT})}{\mathrm{cost}(P^{\lambda})}\right)^{1/\alpha}\right)^d \\
&\leq (2\sqrt{d})^d \cdot k \cdot \left(1 + \frac{(k\cdot\lambda)^{1/\alpha}}{\epsilon}\right)^d \\
&\leq (4\sqrt{d})^d \cdot \left(\frac{k^{\frac{1}{\alpha}+\frac{1}{d}} \cdot \lambda^{1/\alpha}}{\epsilon}\right)^d \\
&\in O\left(\frac{k^{\frac{d}{\alpha}+1}\cdot\lambda^{d/\alpha}}{\epsilon^d}\right)
\end{aligned}$$

where inequality $(*)$ follows from the fact that $r_i \leq \mathrm{cost}(P^{OPT})^{1/\alpha}$. □

We still have to show how to approximate $\mathrm{cost}(P^{OPT})$ for the construction of the coreset S. Feder et al. [FG88] show how to compute deterministically a 2-approximate solution for the so called k-center problem in $O(n\log k)$ time. Furthermore, Har-Peled shows in [Har04] how to obtain such an approximation in $O(n)$ expected time for $k = O(n^{1/3}/\log n)$. The k-center problem differs from the k-disk coverage problem just in the objective function which is given by $\max_{i=1..k}\ r_i^\alpha$ where the discs have radii r_i. Since $\frac{1}{k}\cdot\sum_{i=1}^{k} r_i^\alpha \leq \max_{i=1..k}\ r_i^\alpha \leq \sum_{i=1}^{k} r_i^\alpha$ a 2-approximation for the k-center problem is a $2k$-approximation for the k-disk cover problem. Using such an approximation the size of our coreset becomes $O\left(k^{\frac{2d}{\alpha}+1}/\epsilon^d\right)$. It remains to show how to obtain a solution for the coreset.

2.5.2 Algorithms

Discrete Case

Note that the discrete version of the k-disc cover problem can be solved by the approach of Bilo et. al. [BCKK05]. Recall that their algorithm runs $n^{((\alpha/\epsilon)^{O(d)})}$ time. A short description of this algorithm will be given later in this section.

Alternatively we can find an optimal solution in the following way. We consider all k-subsets of the points in the coreset S as the possible centers of the balls. Note that at least one point in S has to lie on the boundary of each ball in an optimal solution (otherwise you could create a better solution by shrinking a ball). Thus the number of possible radii for each ball is bounded by $n - k$. In total there are $(n-k)^k \cdot \binom{n}{k} \le n^{2k}$ possible solutions. Hence, we have the following theorem:

Theorem 2.20 *The running time of the approximation algorithm in the discrete case is*

$$O\left(n + \left(\frac{k^{\frac{2d}{\alpha}+1}}{\epsilon^d}\right)^{\min\{\,2k,\ (\alpha/\epsilon)^{O(d)}\,\}}\right).$$

Non-discrete Case

Note that on each ball D of an optimal solution there must be at least three points (or two points in diametrical position) that define D - otherwise it would be possible to obtain a smaller solution by shrinking D. Thus for obtaining an optimal solution via exhaustive search it is only necessary to check all k-sets of 3- respectively 2-subsets of S which yields a running time of $O(n^{3k})$. Hence, we have:

Theorem 2.21 *A $(1+\epsilon)$-approximate solution of the non-discrete k-disk cover problem can be found in*

$$O\left(n + \left(\frac{k^{\frac{2d}{\alpha}+1}}{\epsilon^d}\right)^{3k}\right)$$

2.5.3 k-disk Cover with Few Outliers

Assume we want to cover not all points by balls but we relax this constraint and allow a few points not to be covered, i.e. we allow let us say c outliers. This way, the optimal cover might have a considerably lower power consumption/cost.

Conceptually, we think of a k-disk cover with c outliers as a $(k+c)$-disk cover with c balls having radius 0. Doing so, we can use the same coreset construction as above, replacing k by $k+c$. Obviously, the cost of an optimal solution for the $(k+c)$-disk cover problem is a lower bound for the k-disk cover with c outliers. Hence, the imposed grid might be finer than actually needed. So snapping each point to its closest representative still ensures a $(1+\epsilon)$-approximation. When constructed as above, the coreset has size $O\left(\frac{(k+c)^{\frac{2d}{\alpha}+1}}{\epsilon^d}\right)$.

Again, there are two ways to solve this reduced instance, first by a slightly modified version of the algorithm proposed by Bilo et al. [BCKK05] and second by exhaustive search.

We will shortly sketch the algorithm by Bilo et al. [BCKK05] which is based on a hierarchical subdivision scheme proposed by Erlebach et al. in [EJS01] which in turn is based on the works of Hochbaum and Maass [HM85]. Each subdivision is assigned a level and they together form a hierarchy. All possible balls are also assigned levels depending on their size. Each ball of a specific level has about the size of an ϵ-fraction of the size of the cells of the subdivision of same level. Now, a cell in the subdivision of a fixed level is called relevant if at least one input point is covered by one ball of the same level. If a relevant cell Z' is included in a relevant cell Z and no larger cell Z'' exists that would satisfy $Z' \subseteq Z'' \subseteq Z$, then Z' is called a child cell of Z and Z is called the parent of Z'. This naturally defines a tree. It can be shown that a relevant cell has at most a constant number of child cells (the constant only depending on ϵ, α and d). The key ingredient for the algorithm to run in polynomial time is the fact that there exists a nearly optimal solution where a relevant cell can be covered by only a constant number of balls of larger radius. The algorithm then processes all relevant cells of the hierarchical subdivision in a bottom-up fashion using dynamic programming. A table is constructed that for a given cell Z, a given configuration P of balls having higher level than Z (i.e. large balls) and an integer $i \leq k$ stores the balls of level at most the level of Z (i.e. small balls) such that all input points in Z are covered and the total number of balls is at most i. This is done for a cell Z by looking up the entries of the child cells and iterating over all possible ways to distribute the i balls among them.

The k-disk cover problem with c outliers exhibits the same structural properties as the k-disk cover problem without outliers. Especially, the local optimality of the global optimal solution is preserved. Hence, we can adapt the dynamic programming approach of the original algorithm. In order for the algorithm to cope with c outliers we store not only one table for each cell but $c+1$ such tables. Each such table corresponds to the table for a cell Z where $0, 1, \ldots, c$ points are not covered. Now, we do not only iterate over all possible ways to distribute the i balls among its child cells but also all ways to distribute $l \leq c$ outliers. This increases the running time to $n^{((\alpha/\epsilon)^{O(d)})} \cdot c^{((\alpha/\epsilon)^{O(d)})} = n^{((\alpha/\epsilon)^{O(d)})}$. Hence running the algorithm on the coreset yields the following result:

Theorem 2.22 *We can compute a minimum k-disk cover with c outliers* $(1+\epsilon)$ *approximately in time*

$$O\left(n + \left(\frac{(k+c)^{\frac{d}{\alpha}+1}}{\epsilon^d}\right)^{(\alpha/\epsilon)^{O(d)}}\right).$$

For the exhaustive search approach we consider all assignments of k disks each having a representative as its center and one lying on its boundary. For each such assignment we check in time $O(k|S|)$ whether the number of uncovered points is at most c. We output the solution with minimal cost.

Theorem 2.23 *We can compute a minimum k-disk cover with c outliers* $(1+\epsilon)$ *approximately in time*

$$O\left(n + k\left(\frac{(k+c)^{\frac{2d}{\alpha}+1}}{\epsilon^d}\right)^{2k+1}\right).$$

2.6 Information Aggregation via Energy-minimal TSP Tours

While early wireless sensor networks were primarily data collection systems where sensor readings within the network are all transferred to a central computing device for evaluation, current wireless sensor networks perform a lot of the data processing *in-network*. For this purpose some nodes in the network might be interested in periodically *collecting* information from certain other nodes, some nodes might want to *disseminate* information to certain groups of other nodes. A typical approach for data collection and dissemination as well as for data aggregation purposes are tree-like subnetwork topologies, they incur certain disadvantages with respect to load-imbalance as well as non-obliviousness to varying initiators of the data collection or dissemination operation, though. Another, very simple approach could be to have a *virtual token* floating through the network (or part thereof). Sensor nodes can attach data to the token or read data from the token and then hand it over to the next node. Preferably the token should not visit a node again before all other nodes have been visited and this should happen in an energy-optimal fashion, i.e. the sum of the energies to hand over the token to the respective next node should be minimized. Such a scheme has some advantages: first of all none of sensor nodes plays a distinguished role – something that is desirable for a system of homogeneous sensor nodes – furthermore every sensor node can use the same token to initiate its data collection/dissemination operation. Abstractly speaking we are interested in finding a *Travelling Salesman Tour* (TSP) of minimum energy cost for (part of) the network nodes. Unfortunately, the classical TSP with non-metric distance function is even hard to approximate (see [OM]). However, for metrics satisfying the relaxed triangle inequality $\|xy\| \leq \tau(\|xz\| + \|zy\|)$ for $\tau \geq 1$ and every triple of points x, y and z a 4τ approximation exists [BC00] (where $\|xy\|$ denotes the Euclidean distance between point x and z).

In this section we show that the 'normal' Euclidean TSP is not suitable for obtaining an energy-efficient tour and devise a 6-approximation algorithm for TSP under squared Euclidean metric. More generally, we present an $O(1)$-approximation for the TSP problem with powers α of the Euclidean distance as edge weights. For small α we improve upon previous work by Andreae [And01] and Bender and Chekuri [BC00].

2.6.1 Why Euclidean TSP Does Not Work

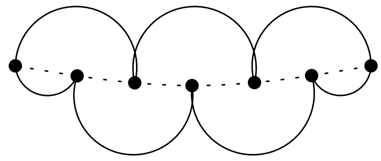

Figure 2.8: An optimal energy-minimal tour for points on a slightly bent line.

Simply computing an optimal tour for the underlying Euclidean instance does not work. The cost for such a tour can be a factor $\Omega(n)$ off from the optimal solution for the energy-minimal tour. Consider the example where n points lie on a slightly bent line and each point having distance 1 to its right and left neighbor. An optimal Euclidean tour would visit the points in their linear order and the go back to the first point. Omitting the fact that the line is slightly bent this tour would have a cost of $(n-1) \cdot 1^2 + (n-1)^2 = n(n-1)$ if the edge weights are squared Euclidean distances. However, an optimal energy-minimal tour would have a cost of $(n-2) \cdot 2^2 + 2 \cdot 1^2 = 4(n-1) + 2$. This tour would first visit every second point on the line and on the way back all remaining points as in Figure 2.8.

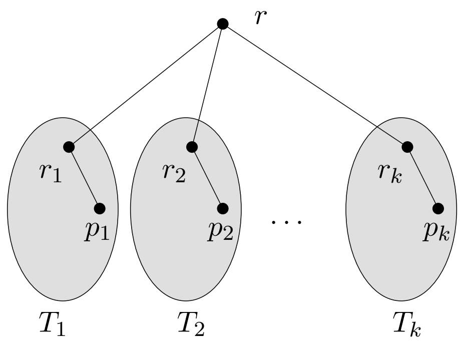

Figure 2.9: Tree T and its children trees $T_1, T_2, \dots, T_k$.

2.6.2 A 6-Approximation Algorithm

In this section we will describe an algorithm which computes a 6-approximation for the TSP under squared Euclidean distance. Obviously, the cost of a minimum spanning tree is a lower bound for the optimal value OPT of the tour. Consider a non-trivial minimum spanning tree T for a graph with node set V and squared Euclidean edge weights. We denote the cost of such a tree by MST(T). Let r be the root of T and p be one child of T.

We define two Hamiltonian paths $\pi^a(T)$ and $\pi^b(T)$ as follows. Let $\pi^a(T)$ be a path starting at r, finishing at p that visits all nodes of T and the cost of this path be at most $6\,\mathrm{MST}(T) - 3\|rp\|^2$. Let $\pi^b(T)$ be defined in the same way but in opposite direction, i.e. it starts at p and finishes at r.

Now, if we have such a tour $\pi^a(T)$ for the original vertex set V we can construct a Hamilton tour by connecting r with p. The cost of this tour is clearly at most $6\,\mathrm{MST}(T) - 3\|rp\|^2 + \|rp\|^2 \le 6\,\mathrm{MST}(T) \le 6\,\mathrm{OPT}$. It remains to show how to construct such tours π^a and π^b. We will do this recursively.

For a tree T of height 1, i.e. a single node r, $\pi^a(T)$ and $\pi^b(T)$ both consist of just the single node. Conceptually, we identify p with r in this case. Obviously, the cost of both paths is trivially at most $6\,\mathrm{MST}(T) - 3\|rp\|^2$.

Now, let T be of height larger than 1 and let $T_1, \dots, T_k$ be its children trees. Let r denote the root of T and r_i the root of T_i and p_i be a child of T_i as in Figure 2.9. Then we set $\pi^a(T) = (r, \pi^b(T_1), \pi^b(T_2), \dots, \pi^b(T_k))$.

The cost of the path $\pi^a(T)$ satisfies

$$\begin{aligned}
\mathrm{cost}(\pi^a(T)) &= \|rp_1\|^2 + \mathrm{cost}(\pi^b(T_1)) + \|r_1p_2\|^2 + \mathrm{cost}(\pi^b(T_2)) + \\
&\qquad \dots + \|r_{k-1}p_k\|^2 + \mathrm{cost}(\pi^b(T_k)) \\
&\le (\|rr_1\| + \|r_1p_1\|)^2 + \mathrm{cost}(\pi^b(T_1)) \\
&\quad +(\|r_1r\| + \|rr_2\| + \|r_2p_2\|)^2 + \mathrm{cost}(\pi^b(T_2)) \\
&\quad \vdots \\
&\quad +(\|r_{k-1}r\| + \|rr_k\| + \|r_kp_k\|)^2 + \mathrm{cost}(\pi^b(T_k)) \\
&\le 2\|rr_1\|^2 + 2\|r_1p_1\|^2 + \mathrm{cost}(\pi^b(T_1)) \\
&\quad +3\|r_1r\|^2 + 3\|rr_2\|^2 + 3\|r_2p_2\|^2 + \mathrm{cost}(\pi^b(T_2)) \\
&\quad \vdots \\
&\quad +3\|r_{k-1}r\|^2 + 3\|rr_k\|^2 + 3\|r_kp_k\|^2 + \mathrm{cost}(\pi^b(T_k)) \\
&\le 6\sum_{i=1}^{k} \|rr_i\|^2 + 3\sum_{i=1}^{k} \|r_ip_i\|^2 + \sum_{i=1}^{k} \mathrm{cost}(\pi^b(T_i)) - 3\|rr_k\|^2 \\
&\le 6\sum_{i=1}^{k} \|rr_i\|^2 + 6\sum_{i=1}^{k} \mathrm{MST}(T_i) - 3\|rr_k\|^2 \\
&= 6\,\mathrm{MST}(T) - 3\|rr_k\|^2.
\end{aligned}$$

In the above calculation we used the fact that $(\sum_{i=1}^n a_i)^\alpha \le n^{\alpha-1} \cdot \sum_{i=1}^n a_i^\alpha$, for $a_i \ge 0$ and $\alpha \ge 1$. This follows directly from Jensen's inequality and the fact that the function $f : x \mapsto x^\alpha$ is convex. The path $\pi^b(T)$ is constructed analogously.

In fact, the very same construction and reasoning can be generalized to the following corollary.

Corollary 2.24 *There exists a $2 \cdot 3^{\alpha-1}$-approximation algorithm for the TSP if the edge weights are Euclidean edge weights to the power α.*

The metric with Euclidean edge weights to the power α satisfies the relaxed triangle inequality with $\tau = 2^{\alpha-1}$. A short computation shows that our algorithm is better than previous algorithms [BC00, And01] for small α, i.e. for $2 \leq \alpha \leq 2.7$.

Conclusions

In this chapter we have addressed the problem of installing a communication infrastructure for a wireless network by suitably assigning transmission ranges to the individual radio stations. We considered several basic tasks that are often performed by wireless networks and we provided algorithms for solving these tasks in an energy-efficient way. Since the problems that we considered are NP-hard, we provided polynomial time approximation schemes and fast algorithms with a constant approximation guarantee. For some problems where polynomial time approximation schemes were already known we could improve the running time drastically upon previous algorithms, like the k-hop broadcast problem or the k-disk cover problem. For other problems we provided the first polynomial time approximation schemes, like the k-set broadcast problem or the k-hop multicast problem. For the travelling salesperson problem under squared Euclidean distance we were able to partially improve the constant approximation guarantee.

All polynomial time approximation schemes that were presented here rely on the fact that we could identify a small coreset for each problem and then solve the problem for this reduced instance. One of the goals of this chapter was to show that the concept of coresets is useful for quite a number of geometric problems and it yields fast approximation algorithms. But coresets are not only the basis of fast algorithms but they are also very general in the sense that they allow the problem to be modified and still one can use the same or a fairly similar coreset to solve the new problem. For instance, we have seen how to find an energy-minimal k-hop broadcast for a set of radio nodes in the plane. One issue, that we did not discuss there was the issue of reliability. There was one path from the source node to each radio node in the wireless network. But what if we would also allow for some radio nodes to fail but still want the source node to broadcast its message to all other nodes within at most k hop. One solution would be to have, lets say, c node-disjoint paths from the source node to each of the other radio nodes. This way $c - 1$ radio nodes could fail and we could still perform the desired broadcast operation. For c being a small constant we could actually use the same coreset construction and solve the problem approximately to any arbitrary relative error. This small example demonstrates the power of our approach that we described here.

For simplicity, we assumed the radio nodes to lie in the Euclidean plane. This is a rather strong assumption and in real world hardly ever true. The metric that is experienced in real world is more general than the Euclidean plane but on the other hand not too general. It still has some resemblance with the Euclidean space. We will show how to deal with this issue in the next chapter. An indicator of the flexibility and robustness of our approach that we used here is demonstrated by the fact that even in this more general metrical setting our algorithms are still able to find $(1 + \epsilon)$-approximate solutions in polynomial time.

Chapter 3

Low-dimensional Metric Spaces

3.1 Introduction

In the last chapter we have seen approximation algorithms for numerous problems that arise in wireless networks. For analytical purposes it was very convenient to assume that all network nodes were placed in the Euclidean plane. Unfortunately, in real-world wireless network deployments, especially if not in the open field, the experienced energy requirement to transmit does not exactly correspond to some power of the Euclidean distance between the respective nodes. Buildings, uneven terrain or interference might affect the transmission characteristics considerably. Nevertheless there is typically still a strong correlation between the actual geographic distance and the required transmission power. An interesting question is now how to model analytically this correlation between geographic distance and energy requirement. One possible way is to assume that the required transmission energies are powers of the distance values in some metric space containing all the network nodes, and that this metric space has some resemblance to a low-dimensional Euclidean space. "Resemblance to a low-dimensional Euclidean space" could be interpreted differently: one might postulate that there is a mapping from this metric space into low-dimensional Euclidean space which more or less preserves distances. This is a rather strong assumption, though. Another means to capture similarity to low-dimensional Euclidean spaces is the so-called *doubling dimension* [GKL03]. The *doubling dimension* of a metric space (X, d), where X is a set of points and d is the distance function, is the least value ρ such that any ball in the metric with arbitrary radius r can be covered by at most 2^ρ balls of radius $r/2$. Note that for any $\rho \in \mathbb{N}$, the Euclidean space $\mathbb{R}^\rho$ has doubling dimension $\Theta(\rho)$.

In this chapter we will consider the doubling dimension a bit more in-depth and give a novel characterization of such metrics based on *hierarchical fat decompositions* (HFDs). We then show how the algorithms for wireless communication problems that were presented in the previous chapter as well as other algorithms in the wireless networking context can be adapted to arbitrary metric spaces of bounded doubling dimension. Interestingly, we could also show that metrics of bounded doubling dimension are not a tight characterization of all the metrics that allow for well-behaved HFDs, that is, there are metrics which are *not* of bounded doubling dimension, but still our and many other algorithms run efficiently. Finally, in Section 3.5 we examine metrics of bounded doubling dimension that arise as shortest-path metrics in unweighted graphs (e.g. unit-disk communication graphs). We show that for such metrics, an HFD can be computed in near-linear time, and the latter can be instrumented to derive a simple deterministic routing scheme that allows for $(1+\epsilon)$ stretch using routing tables of size $O\left(\left(\frac{1}{\epsilon}\right)^{O(\rho)} \cdot \log^2 n\right)$ bits.

Related Work

Metrics of bounded doubling dimension have been studied for quite some time, amongst others Talwar in [Tal04] provides algorithms for low-dimensional metrics that $(1+\epsilon)$ approximate various optimization problems like TSP, k-median, and facility location. Furthermore he gives a construction of a well-separated pair decomposition for unweighted graphs of bounded doubling dimension ρ that has size $O(s^\rho n \log n)$ (for doubling constant s). Based on that he provides compact representation schemes like approximate distance labels, a shortest path oracle, as well as a routing scheme which allows for $(1+\epsilon)$-paths using routing tables of size $O\left(\left(\frac{\log n}{\epsilon}\right)^\rho \log^2 n\right)$. [1] An improved routing scheme using routing tables of size $O\left((1/\epsilon)^{O(\rho)} \log^2 n\right)$ bits was presented in [CGMZ05] by Chan et al., but the construction is rather involved and based on a derandomization of the Lovasz Local Lemma. Har-Peled and Mendel in [HM06] gave a randomized construction for a well-separated pair decomposition of *linear* size which matches the optimal size for the Euclidean case from Callahan and Kosaraju in [CK95].

Our Contribution

First, we give a novel characterization of metrics of bounded doubling dimension that is rather straightforward and intuitive using hierarchical fat decompositions (HFDs). As a side result we show how such HFDs directly lead to well-separated pair decompositions of linear-size for metrics of bounded doubling dimension (such WSPDs were also constructed in a randomized fashion in [HM06]). Second, we show how the algorithms for wireless communication problems that were presented in the last chapter can be made run efficiently when the network is placed in a metric of bounded doubling dimension. Furthermore

[1] In fact his results are more general in a sense that they hold for metrics with arbitrary spread, but all constants depend on the spread.

we show that metrics of bounded doubling dimension are not a tight characterization of all the metrics that allow for well behaved HFDs, that is there are metrics which are not of bounded doubling dimension but still our algorithms for wireless communication problems run in polynomial time. And last, we show how to construct HFDs efficiently in unweighted graphs as they occur for example as communication graphs in networks and instrument the HFDs to allow for a routing scheme that guarantees paths at most a $(1+\epsilon)$ factor longer than the shortest paths using routing tables of size $O\left(\left(\frac{1}{\epsilon}\right)^{O(\rho)} \cdot \log^2 n\right)$ bits. Our construction is considerably simpler than the one given in [CGMZ05] by Chan et al.

3.2 Properties of Low-dimensional Metrics

As mentioned in the introduction, the theoretical analysis of algorithms typically requires some simplifying assumptions on the problem setting. In case of wireless networking, a very common assumption is that all the network nodes are in the Euclidean plane, distances are the natural Euclidean distances, and the required transmission energy is some power of the Euclidean distance. This might be true for network deployments in the open field, but as soon as there are buildings, uneven terrain or interference, the effective required transmission power might be far higher. Still, it is true that there is a strong correlation between geographic/Euclidean distance and required transmission power. One way to capture similarity to low-dimensional Euclidean spaces is the so-called *doubling dimension*. The *doubling dimension* of a metric space (X, d) is the least value ρ such that any ball in the metric with arbitrary radius r can be covered by at most 2^ρ balls of radius $r/2$. In the following we will consider such metrics of bounded doubling dimension, i.e. the metrics whose doubling dimension is a constant. In particular, we will show that a bounded doubling dimension does not only imply a Euclidean-like covering property but also a *packing* property.

3.2.1 Metrics of Bounded Doubling Dimension

The fact that every ball can be covered by at most a constant number of balls of half the radius (covering property) induces the fact, that not too many balls of sufficiently large radius can be placed inside a larger ball (packing property). The following lemma states this fact precisely. (The same observation was made in section 2 of [HM06] in the context of net-trees but was not explicitly stated in this general form.)

Lemma 3.1 (Packing Lemma) *Given a metric (X, d) with doubling constant k, i.e. every ball can be covered by at most k balls of half the radius,* *then, at most k pairwise disjoint balls of radius $r/2+\epsilon$, for $\epsilon > 0$ can be placed inside a ball of radius r.*

Proof: Consider a ball B of radius r. Place a set $S = \{B_1, B_2, \ldots, B_l\}$ of pairwise disjoint balls each having radius $r/2+\epsilon$ inside B. Let $C = \{b_1, b_2, \ldots, b_k\}$ be a set of balls of radius $r/2$ that cover the ball B. The distance between two centers of balls from S is at least $r+2\epsilon > r$ as they are pairwise disjoint. Hence, every ball $b_i \in C$ can cover at most one center of a ball $B_j \in S$. Since every ball from the set S is covered and especially its center, we have $|S| \leq |C| = k$. □

The same generalizes to arbitrary radii. If a ball B of radius r can be covered by at most k balls of radius r then there can be at most k pairwise disjoint balls of radius $r+\epsilon$ for $\epsilon > 0$ placed inside B. We will make use of this packing property at various places later.

3.2.2 Hierarchical Fat Decompositions (HFD)

Given an arbitrary metric (X, d), a *decomposition* is a partition of X into clusters $\{C_i\}$. A *hierarchical decomposition* is a sequence of decompositions $P_l, P_{l-1}, \ldots, P_0$, where each cluster in P_i is the union of clusters from P_{i-1}, $P_l = X$, and $P_0 = \{\{x\}|x \in X\}$, i.e. P_l is the single cluster containing X and every point forms one separate cluster in P_0.[2] We refer to clusters of P_i as clusters at level i. A hierarchical decomposition where each cluster of the same level i is contained in a ball of radius r_i, contains a ball of radius $\alpha \cdot r_i$, and $r_{i-1} \leq \beta \cdot r_i$ for constants α and $\beta < 1$ is called a *hierarchical fat decomposition* (HFD). Thus, in an HFD clusters are fat and the size of the clusters from different levels form a geometric sequence. We call a set *fat* if the ratio between an inscribed ball and a surrounding ball is bounded by a constant.

We will show how to construct an HFD for an arbitrary metric (X, d). Without loss of generality we assume $\min_{p,q\in X} d(p, q) = 1$. We call $\Phi = \max_{p,q\in X} d(p, q)$ the *spread* of X. We construct the HFD bottom-up. Let L_i be a set of points which we call landmarks of level i. With each landmark we associate a cluster $C_i(l) \subseteq X$. Figure 3.1 illustrates the construction.

On the lowest level we have $L_o = X$ and $C_0(l) = \{l\}$, i.e. each point forms a separate cluster. Obviously, each cluster is contained in a ball of radius 1 and contains a ball of radius $\frac{1}{2}$.

Starting from the lowest level we construct the next level recursively as follows. For level i we compute a 4^i-independent maximal set (i.e. a maximal set with respect to insertion with the pairwise distance of at least 4^i) of landmarks L_i from the set L_{i-1} of landmarks from one level below. Hence, the distance

[2] This is also known as a laminar set system as used frequently in the literature.

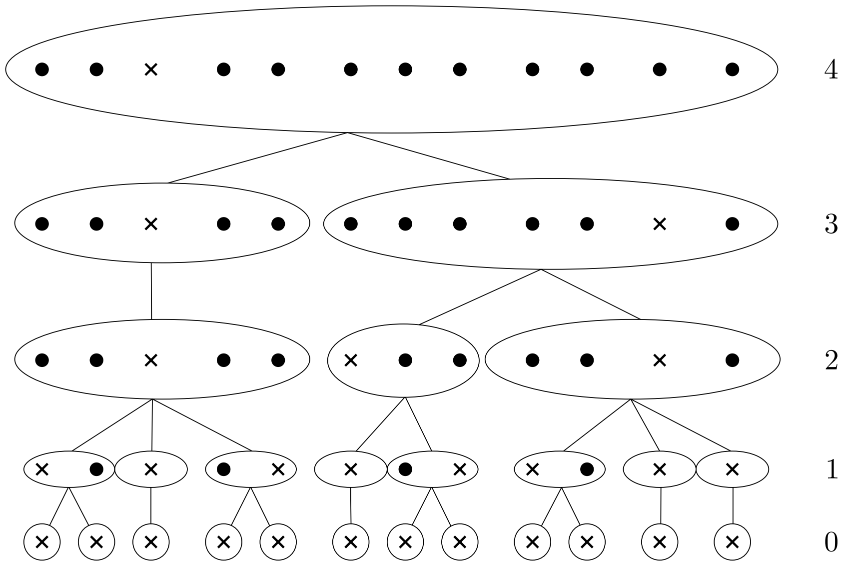

Figure 3.1: A hierarchical fat decomposition (HFD) for a point set is drawn here schematically. All five different levels from level 0 to level 4 are displayed. Each cluster of each level is represented by an ellipse and each landmark of each cluster is marked by a cross. An HFD naturally defines a tree.

between any two landmarks of level i is at least 4^i. We compute the Voronoi diagram VD of this set L_i and call the Voronoi cell of l $VC_i(l)$. The union of all clusters of landmarks from level $i-1$ that fall in the region $VC_i(l)$ form the new cluster that we associate with landmark l, i.e. $C_i(l) = \bigcup_{p \in VC_i(l)} C_{i-1}(p)$. Obviously, each Voronoi cell contains a ball of radius $4^i/2$ and is contained in a ball of radius 4^i, since the set of landmarks L_i form a 4^i maximal independent set. Hence, each cluster on level i is contained in a ball of radius $\sum_{j=0}^{i} 4^j \leq 4^{i+1}/3$ and each cluster contains a ball of radius $4^i/2 - \sum_{j=0}^{i-1} 4^i \geq 4^i/6$. Thus, we have constructed an HFD.

3.2.3 A Characterization of Metrics of Bounded Doubling Dimension

We say an HFD has degree d if the tree induced by the hierarchy has maximal degree d. The following theorem gives a characterization of metrics with bounded doubling dimension in terms of such HFDs.

Theorem 3.2 *A metric (X, d) has bounded doubling dimension if and only if all hierarchical fat decompositions of (X, d) have bounded degree.*

Proof: First, suppose metric (X, d) has bounded doubling dimension. Fix an arbitrary HFD for (X, d) and pick a cluster C. Since C is fat, it is contained

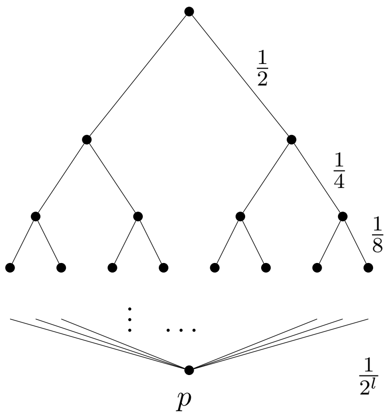

Figure 3.2: A metric with unbounded doubling dimension but with bounded degree HFD.

in a ball of radius r_1 and it is the union of fat clusters $\{C_1, C_2, \ldots, C_l\}$. Each of them contains a ball of radius r_2. The ratio of the two radii r_1 and r_2 is bounded by a constant due to the definition of an HFD. Then, by the Packing Lemma 3.1 cluster C cannot contain more than a constant number of clusters from the level below. Hence, each HFD has bounded degree.

On the other hand, suppose (X, d) has no bounded degree. Then there exists a ball $B(x, r) = \{y|d(x, y) \leq r\}$ that cannot be covered by a constant number of balls of half the radius r. We can construct an HFD, which has no bounded degree as follows. Consider an HFD constructed as in Section 3.2.2, where the set of landmarks always contains the point x. Consider the minimal cluster C that contains ball $B(x, r)$ and consider the set of children clusters $\{C_1, C_2, \ldots, C_l\}$ of C that are all contained in a ball of radius $r/2$. Due to the definition of an HFD the difference in the levels of these clusters is bounded by a constant. Since, the number of children clusters is not bounded, the HFD cannot have bounded degree. □

There are metrics however, that admit an HFD with bounded degree but do not have bounded doubling dimension, i.e. the set of metrics with bounded doubling dimension are a proper subset of metrics that admit an HFD with bounded degree. The following metric is such an example. Consider the complete binary tree of depth l and each edge from level $i-1$ to level i having weight $\frac{1}{2^i}$ as in Figure 3.2. Let p be a node which is connected to all leaves with edge weights $\frac{1}{2^l}$. The shortest path metric induced by this graph does not have a bounded doubling dimension but admits an HFD with bounded degree. We can place 2^l disjoint balls of radius $\frac{1}{2^{l+1}}$, each having a leaf as its center, inside a ball of radius $\frac{1}{2^l}$ with center p. Hence, the metric cannot have bounded doubling dimension for arbitrary large l (Packing Lemma). On the other hand, it is easy to see that the metric has an HFD of degree 2.

3.3 Well-separated Pair Decomposition

Suppose we are given an arbitrary metric (X, d), where X is a set of n points and $d : X \times X \mapsto \mathbb{R}_{\geq 0}$ is the distance function. To store the distance function we need $\Theta(n^2)$ space. However, what happens if we do not need to store the exact values of the distance function but, lets say, we allow a small error of $1/s$. There is some hope that we could then store the complete distance matrix in less than $\Theta(n^2)$ space. In fact, it was shown by Callahan and Kosaraju [CK95] that for Euclidean spaces of fixed dimension we can store all pairwise distances approximately with a relative error of $1/s$ in $\Theta(n)$ space for fixed s. Unfortunately, for arbitrary metric spaces it is in general not possible to store all pairwise distances even approximately in less than $\Theta(n^2)$ space. However, in this section we will show that for metrics of bounded doubling we can store all pairwise distances with a relative error of $1/s$ in $\Theta(n)$ space. This matches the bound for the Euclidean case.

Talwar [Tal04] gave a randomized construction for metrics of bounded doubling dimension but the space required to store the distance function still depended on the spread of the metric. Later, Har-Peled and Mendel [HM06] also gave a randomized construction where they removed the dependency on the spread. We will see that with our hierarchical fat decomposition a deterministic construction without the dependency on the spread follows immediately.

So lets assume we have a metric space (X, d), and let $A \subseteq X$ and $B \subseteq X$ be two subsets of points. We say that the point sets A and B are *well-separated* if A and B can each be contained in a ball of radius r whose minimum distance is at least $s \cdot r$, where s is the *separation* and assumed to be a constant. Let us define the *interaction product*, denoted by $\otimes$, between two point sets A and B as $A \otimes B = \{\{p, p'\} \mid p \in A,\, p' \in B,\, p \neq p'\}$. The set $\{\{A_1, B_1\}, \ldots, \{A_k, B_k\}\}$ is said to be a *realization* of $A \otimes B$ if

1. $A_i \subseteq A$ and $B_i \subseteq B$ for all $i = 1, 2, \ldots, k$.
2. $A_i \cap B_i = \emptyset$ for all $i = 1, 2, \ldots, k$.
3. $(A_i \otimes B_i) \cap (A_j \otimes B_j) = \emptyset$ for all $1 \leq i < j \leq k$.
4. $A \otimes B = \bigcup_{i=1}^{k} A_i \otimes B_i$.

The realization is said to be *well-separated* if it additional satisfies the property: A_i and B_i are well-separated for all $i = 1, 2, \ldots, k$.

Let T be a tree associated with X, that is each of the leafs of T is labeled by a singleton set containing one of the points of X and each internal node is labeled by the union f all sets labeling the leaves of its subtree. For $A, B \subseteq X$ we say that a realization of $A \otimes B$ uses T if all A_i and B_i in the realization are nodes in T. We define a *well-separated pair decomposition (WSPD)* of X

to be a structure consisting of a tree T associated with X and a well-separated realization of $X \otimes X$ that uses T.

Once we have given a well-separated pair decomposition for a point set X and a separation constant s we can simply read of the approximate distance for any two points $p \in X$ and $p' \in X$ and the approximate distance has a relative error of $1/s$ compared to the true distance between p and p'.

We have the following theorem:

Theorem 3.3 *An HFD with bounded degree immediately implies a well-separated pair decomposition (WSPD) of linear size in the number of input points for metrics of bounded doubling dimension.*

Proof: The construction follows closely the lines of [CK95]. If we replace in their construction the *fair split tree* by our hierarchical fat decomposition, we get the same bounds, apart from constant factors. All we need to show is that lemma 4.1 in [CK95] still holds, i.e. if a ball B of radius r is intersected by the surrounding balls of a set of clusters $S = \{C_1, C_2, \ldots, C_l\}$ with $C_j \cap C_j = \emptyset$ for $i \neq j$ and the parent of each cluster C_i has a surrounding ball of radius larger than r/s for a constant s, then the set S can only contain a constant number of clusters. But this is certainly true. The Packing Lemma 3.1 assures that there are just a constant number of clusters whose surrounding balls intersect a large ball B whose radius is larger by a constant. And as the HFD has bounded degree, these clusters have constant number of children clusters $S = \{C_1, C_2, \ldots, C_l\}$ all together. If we eliminate all clusters in the HFD that just have one children cluster we get that the number of well-separated pairs is linear in the number of input points and depends only on the constant s and the doubling dimension. □

3.4 Optimizing Energy-efficiency in Low-dimensional Metrics

In the following we will briefly sketch how the algorithms presented in the last chapter for wireless communication problems can also be applied for metrics of bounded doubling dimension. Furthermore we show how an old result ([FMS03]) can also be partly adapted from the Euclidean setting.

3.4.1 Energy-efficient k-hop Broadcast, k-set Broadcast and k-disk cover

The algorithm presented in Section 2.2 for broadcasting in the plane can be generalized to arbitrary metrics with bounded doubling dimension.

Obviously, the Structure Lemma 2.2 still holds since the triangle inequality holds. Now, instead of placing a grid onto the plane, we construct an HFD for the nodes as in Section 3.2.2. The level of the decomposition where each cluster is contained in a ball of radius $r = \Delta/2$ replaces the grid in the approximation algorithm. As the metric has bounded doubling dimension, the HFD has bounded degree. Hence, there is just a constant number of clusters in the decomposition of this level. We can solve this instance in the same way as for the planar case. The construction of the decomposition can be done in a naive way in time $O(n^2)$, but we believe that a faster construction should be possible using techniques similar to those used in [HM06]. We want to emphasize that our algorithm essentially works for *any* metric, but for metrics of bounded doubling dimension we can guarantee polynomial running time. In fact, our algorithm runs in polynomial time if we have constructed an HFD of bounded degree. This is certainly true for metrics of bounded doubling dimension but as we have seen there exist metric that admit an HFD of bounded degree but do not have a bounded doubling dimension. Even for these metrics, our algorithm runs in polynomial time.

A close look at the approximation algorithms from the previous chapter for energy-efficient k-set broadcast problem and the k-disk cover problem reveals the same properties as for the k-hop broadcast problem. Again, the whole input can be assumed to be in a ball of radius 1 and the grid is replaced by a the appropriate level of the HFD. The number of clusters in this level only depends on k, ϵ and the doubling dimension and hence can be thought of as being constant. Thus, the coreset again has constant size and we solve it in the same way in the planar case.

3.4.2 Energy-efficient k-hop Paths

In [FMS03] the authors have considered the problem of computing an $(1+\epsilon)$ energy-optimal path between a given source node s and a target node t in a network in $\mathbb{R}^2$ which uses at most k hops/transmissions. Again, the assumption was that the required energy to transmit a message from some node p to some other node q is $\|pq\|^\alpha$, for $\alpha \geq 2$ where $\|pq\|$ denotes the Euclidean distance. Using a rather simple construction where the neighborhood of the query pair s and t was covered using a constant number of grid cells (depending only on k, δ, ϵ) such queries could be answered with a $(1+\epsilon)$ guarantee in $O(\log n)$ time. Similarly to the bounded-hop broadcast, we can replace this grid by a respective level of an HFD. For bounded doubling dimension we then know that there are only a constant number of relevant grid cells and the algorithm can be implemented as in the Euclidean case. In [FMS03] the construction was further refined by using a well-separated pair decomposition to actually precompute a linear number of k-hop paths which then for a query could be accessed in $O(1)$ time (independent of k, δ, ϵ). We have not investigated whether this construction also translates to metrics of bounded doubling dimension (where a linear-size WSPD exists).

3.5 Computing HFDs in Shortest-path Metrics

In wireless sensor networks, typically the employed network nodes are very low-capability devices with simple computing and networking units. In particular, most of these devices do not have the ability to adjust the transmission power but always send within a fixed range. The graph representing the pairs of nodes that can communicate with each other is then a so-called *unit-disk graph* (UDG), where two nodes can exchange messages directly iff they are at distance of most 1. Typically UDGs are considered in the Euclidean setting, but they can be looked at in any metric space. Due to the fixed transmission power saving energy directly by varying the latter is not possible. Still, indirectly, energy can be saved by for example better routing schemes which yield to shorter (i.e. fewer hops) paths. In the following we will briefly discuss how HFDs can be used to provide such efficient routing schemes. We first show how in case of unweighted graphs like UDGs, HFDs can be efficiently computed and then we sketch how the structure of the HFDs can be exploited to allow for a routing scheme with almost optimal path lengths using only small routing tables at each node.

3.5.1 A Near-linear Time Algorithm

Consider an unweighted graph $G = (V, E)$. All shortest paths define a shortest-path metric on the set of vertices. If the metric has bounded doubling dimension we can construct an HFD with bounded degree. We describe here how to do this efficiently.

We follow the generic approach described in Section 3.2.2. At level i we need to construct an 4^i-independent maximal set of nodes L_i, the landmarks. This can be done greedily using a modified breadth-first search algorithm on the original graph G. At the same time we can compute the corresponding Voronoi diagram.

We pick an arbitrary node n_1 and add it to the set L_i. In a breadth-first search we successively compute the set of nodes that have distance 1, that have distance 2, and so on until we computed the set of nodes at distance 4^i. We mark each visited node as part of the Voronoi cell of node n_1 and store its distance to n_1. From the set of nodes at distance 4^i we pick a node n_2 and add it to L_i. Starting from node n_2 we again compute the set of nodes that have distance 1, distance 2, and so on to the node n_2. Similarly, if a node is not assigned to a Voronoi cell, we assign it to n_2. If it has been assigned already to some other node but the distance to the other landmark is larger than to the current node n_2, we reassign it to the current node. We do this until no new landmark can be found and all nodes are assigned to its Voronoi cell.

It happens that we visit a node and an edge several times. However, as the metric has bounded doubling dimension, we visit each edge and node at most a constant number. For any edge or node there are just a constant number of landmarks within distance 4^i (cf. Packing Lemma 3.1). Hence, it is visited only a constant number. Thus, the running time is $O(m+n)$ for one level and $O((m+n)\log n)$ for the whole construction of the HFD as there are $O(\log n)$ levels.

3.5.2 Hierarchical Routing in Doubling Metrics

The HFD constructed above implicitly induces a hierarchical naming scheme for all nodes of the network by building IP-type addresses which reflect in which child cluster of each level a node v is contained (remember that there are always only a constant number of children of each cluster). For example if v is contained in the top-most cluster 4, in the 2nd child of that top-most cluster and in the 5th child of that child, its name would be 4.2.5. Clusters can be named accordingly and will be prefixes of the node names. We now install routing tables at each node which allow for almost-shortest path routing in the network: For every cluster $\mathcal{C}$ with diameter D we store at all nodes in the network which have distance at most $O(D/\epsilon)$ from $\mathcal{C}$ a distance value (associated with the respective address of the cluster and a pointer to the predecessor on the shortest path to the cluster) to the boundary of $\mathcal{C}$ in the node's routing table. Now, when a message needs to be routed to a target node t and is currently at node p, p inspects its routing table and looks for an entry which is a as large as possible prefix of the target address. p then forwards the message to the adjacent neighbor which is associated with this routing table entry. A simple calculation shows that this yields paths which are at most a $(1+\epsilon)$ factor longer than the optimal shortest path [3]. For the size of the routing table first consider an arbitrary node v and clusters of diameter at most D. Clearly there are at most $O((1/\epsilon)^{O(\alpha)})$ many such clusters which have distance less than $O(D/\epsilon)$ from v and have hence created a routing table entry at v. Overall there are only $\log n$ levels and each routing table entry has size $O(\log n)$ (since the maximum distance is n). Hence the overall size of the routing table of one node is $O((1/\epsilon)^{O(\alpha)}\log^2 n)$.

3.6 Conclusions

We have seen how to generalize algorithms for problems that lie in Euclidean space to metric spaces with bounded doubling dimension. These metrics are much more general than Euclidean spaces and are thus able to capture real world instances much better. Still, our algorithms for energy-efficient wireless communication that were based on coresets run efficiently for such metrics. The core idea that we used was our construction of hierarchical fat decompositions

[3] In [FGNW06] a similar construction was used to get paths which are at most a constant factor longer than the shortest path. The proof follows along the same lines.

(HFD). It was sufficient for the algorithms that the metric allows for a bounded degree HFD, which is certainly the case for metrics of bounded doubling dimension. We have seen that an HFD also immediately implies a well-separated pair decomposition for metrics of bounded doubling dimension. Furthermore, regarding our characterization of metrics of bounded doubling dimension, it turns out that there exist metrics which do not have bounded doubling dimension but still, a bounded degree HFD exists. It might be interesting to find a tighter characterization of such metrics which allow a bounded degree HFD.

Chapter 4

k-hop Minimum Spanning Trees in Euclidean Metrics

4.1 Introduction

We are given set P of n points in d-dimensional Euclidean space with the distance function $d(\cdot)$, a fixed positive integer k and a root node $r \in P$. The k-hop spanning tree of P is a tree T rooted at r and spanning all points of P, such that number of edges on any root-leaf path is not greater than k. The cost of T is the sum of its edge weights. In this paper we consider the k-hop spanning tree problem of minimum cost (k-hop MST). Figure 4.1 shows an example of a 2-hop spanning tree.

Based on the methods of Arora et al. [ARR98] for the Euclidean k-median problem, we present a polynomial-time approximation scheme for the k-hop MST problem in the plane, when k is a constant.

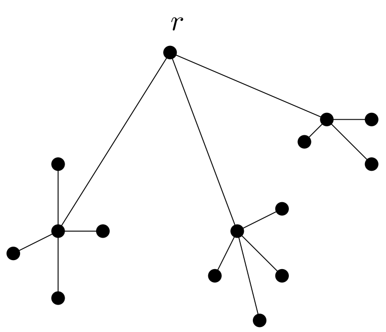

Figure 4.1: A 2-hop minimum spanning tree. All root-leaf paths contain at most 2 edges.

As a byproduct of our algorithm, we also provide a polynomial-time approximation scheme for the geometric versions of the following more general problems:

The multi-level concentrator location problem. Here, we are given a set P of nodes, a set $C \subset P$ of clients and a k sets of facilities $F = F_1 \cup \ldots \cup F_k \subseteq P$ with the opening facility costs f_j for each facility $j \in F$. The task is to open subsets of facilities $F'_i \subseteq F_i$, $1 \leq i \leq k$ and assign each client to the closest level one facility in F'_1, and assign each of the level $(i-1)$ facilities to the closest level i facilities F'_i, such that the opening facilities costs plus the sum of the distances is minimized.

The bounded-depth minimum Steiner tree problem. Given a set P of nodes, a set D of Steiner points and a root node $r \in P$, the task is to construct a minimum cost tree of depth k, rooted at r that spans the set P and possibly uses some Steiner points from the set D.

It is not difficult to see that the k-hop MST is just a special case of the above two problems, and any solutions for them would immediately imply a solution for the k-hop MST problem.

Motivation

Minimum-cost spanning trees are pervasive and their efficient construction appears important in many practical applications. For example, in *multicast-routing* problem in the area of computer networks (see, e.g. [DC90, DEF$^+$94]) a number of clients and a server are connected by a common communication network. The server wishes to transmit identical information to all client nodes. Most solutions to the multicast problem involve computing a tree rooted at the server and spanning the client nodes. The server then transmits the data to its immediate children in the tree and intermediate nodes forward incoming data to their respective descendants in the tree. Tree-routing schemes allow for fast data delivery while keeping the total network load low. Kompella et al. [KPP93] consider the problem of computing multicast-trees that minimize the overall network cost as well as the maximum transmission latency on any path in the tree connecting the server to a client node. It is not hard to see that a multi-hop transmission with too many hops will increase the latency of the communication. Moreover, transmission with too many hops will inevitably increase the probability of a link failure. Thus, assuming that all links in the network have roughly the same transmission delay (which is a reasonable assumption in local area networks), limiting the number of hops in the transmission to some small integer k helps in achieving *fast* and *reliable* communication protocols.

Related Work

In the classic *metric facility location problem*, we are given a set of clients C and a set of facilities F with metric edge costs c_{ij}, for all $i \in F, j \in C$ and opening cost f_i for all facilities $i \in F$. The goal is to open subset of facilities $F' \subseteq F$ such that the sum of opening facility costs, plus the sum of the costs of assigning each client to its closest facility in F' is minimized. The best known approximation algorithm is by Mahdian et al. [MYZ02] that achieves 1.52 approximation ratio. Note that the 2-hop MST is a special case of the facility location problem, e.g. replace each facility cost f_i by the distance from i to the root r. Thus, all the approximation results for facility location problem apply immediately to the 2-hop MST. Guha and Khuller [GK99] proved that the existence of a polynomial time 1.463-approximation algorithm for the metric facility location problem would imply that P = NP. This hardness result also applies for the 2-hop MST problem.

For the Euclidean facility location problem a randomized PTAS based on Arora's technique [Aro98] for the Euclidean TSP is presented in [ARR98], for the points in the plane. Unfortunately, for d-dimensional geometric instances and $d > 2$, the algorithms runs only in quasi-polynomial time. However, Kolliopoulos and Rao [KR99] were able to construct a nearly linear time randomized PTAS for facility location problem for any d-dimensional Euclidean space. The small errors in this paper were fixed by the authors in [KR07].

Zhang [Zha04] gives a 1.77-approximation algorithm for the metric *two-level concentrator location problem* which is a generalization of the 3-hop MST.

The first constant factor approximation for the bounded depth steiner tree problem and likewise for the k-hop MST in general metric spaces is presented by Kantor and Peleg in [KP06]. The approximation ratio of their algorithm is rather high, though. More precisely, they construct a polynomial time approximation algorithm with approximation ratio $1.52 \cdot 9^{k-2}$ for complete graphs whose weight function is a metric.

Althaus et al. [AFHP$^+$05] present an approximation algorithm that computes a k-hop spanning tree in general metric spaces of total expected cost $O(\log n)$ times the cost of the optimal k-hop MST. They approximate the metric space into a tree metric using the result by Fakcharoenphol et al. [FRT03] who showed that any metric space can be probabilistically approximated by a family of tree metrics such that the expected stretch in the cost is at most $O(\log n)$. Althaus et al. develop an exact algorithm for the k-hop MST in the special case when the cost function is induced by a tree.

Clementi et al. [CIM$^+$05] present an algorithm that computes with high probability a constant approximation for constant k for random instances in the plane.

Our Contribution

In this chapter we present the first PTAS for the k-hop MST problem in the plane. We extend the technique of Arora et al. [ARR98] for the Euclidean k-median problem and show that the $(1+\epsilon)$ solution for the k-hop MST problem can be computed in polynomial time.

In Section 4.2 we review the quadtree dissection from [ARR98] and show that there exists a $(1+\epsilon)$ solution to the k-hop MST problem with respect to the given dissection. Furthermore, in Section 4.3 we show how to compute such an approximate solution with a dynamic programming algorithm in time $\left(\frac{n}{\epsilon}\right)^{O(k/\epsilon)}$.

We also extend our algorithm to *the multi-level concentrator location problem* and *the bounded depth minimum Steiner tree problem* in Section 4.4.

4.2 Preliminaries

In this part we describe the quadtree dissection from [ARR98] and show the existence of approximately optimal solutions with a simple structure based on a given dissection. Let P denote a set of n points in the plane. The *bounding box* is the smallest axis-aligned square that contains all points of P. The side-length of the bounding box is the length of its side. In the following, we assume that the bounding box of the points has side-length $L = n/\epsilon$ and all points of P lie on gridpoints of the unit grid defined on the bounding box. Note that the cost increase of the optimum is negligible since moving each point to the closest grid point will increase the minimum cost k-hop MST by at most $\epsilon \cdot OPT$.

A *dissection* of a box is a recursive partition of the box into lower level boxes. More precisely, we view the dissection as a hierarchical decomposition of the plane into boxes. A box in a dissection is any box that can be obtained by a recursive splitting process that starts with the bounding box and generally splits an existing dissection box by 2 axis-orthogonal lines passing through its center into 4 identical subboxes. Such a decomposition naturally defines a 4-ary tree. Each line is assigned a level. There are 2^i level i lines that partition level i boxes into level $i+1$ boxes. The *size* of a box is its side length. A nice property of the dissection boxes is that any two boxes either have disjoint interiors or one is contained inside the other. Note that there are $O(L^2)$ nodes in the tree and its depth is $\log L = O(\log(n/\epsilon))$. The dissection is depicted in Figure 4.2. This dissection is quite similar to our hierarchical fat decomposition (HFD) developed in Chapter 3.

We randomize the levels in the dissection of the bounding box the same way as in [Aro98, ARR98]. Namely, randomly pick two integers $0 \leq a, b < L$. The (a, b)-shift of the dissection is defined by shifting x and y coordinates of all lines by a and b respectively, and then reducing modulo L.

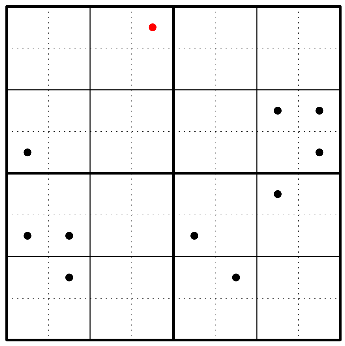

Figure 4.2: The bounding box and the dissection of the plane with a point set is displayed. The thick lines correspond to level 0 lines, the thin lines correspond to level 1 lines and the dotted lines are level 2 lines.

Note that the solution to the k-hop MST problem consists of a collection of line segments. We will only allow the segments to bend and pass through a set of prespecified points called *portals*. More precisely, place $2^i m$ equally spaced portals on each level i line. Moreover, at the corner of each dissection box place a portal. Note that each level $i+1$ box in the dissection has m portals on its two level $i+1$ edges and strictly less than m portals on its two level i edges. In general, any box in the dissection has at most $4m$ portals.

A solution to the k-hop MST problem is called *portal-respecting* if it crosses a dissection box only at portals.

Suppose we are given the optimal set of line segments that describe an optimal k-hop MST solution. To make the solution portal-respecting, we need to deflect each edge that crosses a side of a box in the dissection to the nearest portal. Note that if the size of the box is l, we need to deflect each edge by at most l/m to make it pass through a portal.

Since the shifts a and b are chosen randomly, we have that the probability that each vertical/horizontal line l in the grid is from the level i: $Pr[l \text{ is at level } i] = 2^i/L$. Using this fact, Arora at el. [ARR98] show the following Structure theorem:

Lemma 4.1 *For any collection of line segments, random shifts a and b and $m \geq 1$, the bending process will, with probability at least 1/2, deflect the segments by at most $O(\log L/m)$ times the sum of the length of the line segments.*

Since the above Lemma 4.1 holds for any set of line segments, it also implies the following:

Corollary 4.2 *Let $r \in P$ denote the root node and let shifts a and b be chosen* *uniformly at random. Let* $m = O\left(\frac{\log(n/\epsilon)}{\epsilon}\right)$ *for any* $\epsilon > 0$*. Then, with probability of at least* $1/2$ *the cost of the optimal portal-respecting solution for the* k*-hop MST problem is at most* $(1+\epsilon) \cdot OPT$*, where* OPT *denotes the optimal cost of the* k*-hop MST.*

4.3 The Algorithm

In this section we will describe the algorithm to compute an optimal portal-respecting k-hop MST which is, with probability of at least $1/2$, a $(1+\epsilon)$-approximation to the optimal k-hop MST.

Consider any optimal k-hop MST. We assign each node a level depending on the number of hops to the root r, where r is assigned level 0, its immediate neighbors are assigned 1 and so on. We also assign levels to the edges. An edge from a level $i-1$ node to a level i node is assigned the level i. Hence, we have nodes from level 0 to k and edges from level 1 to k.

Consider now a box in the dissection as described in the previous section. Remember that edges are only allowed to cross the boundary of the box at portals. The optimal solution inside the box is fully determined if we know for each portal and each level i the distance from the portal to the nearest node of level i outside the box. Conversely, the optimal solution outside this box is fully determined if we know for each portal and each level i the distance from the portal to the nearest node of level i inside this box.

Hence, if we fix all distances at the portals of a box to all nearest nodes of levels 0 to $k-1$, only the solution inside this box with minimal cost can be part of an optimal solution. This enables us to design the following dynamic program.

We store in the table $\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$ the cheapest solution for box B that respects the given inside and outside function, where inside_i denotes the distance function on the portals to the closest node of level i inside box B. Function outside_i is defined analogously. In other words inside_i describes what box B can provide to the outside and outside_i describes what can be provided to box B from outside. For the distance function inside_i we can still allow an additional additive error of l/m as the distance between two neighboring portals is already l/m. Remember that the size of box B is l and we have placed m portals on its boundary. Thus, we have $\text{inside}_i(p) \in \{0, l/m, 2l/m, \ldots, 2l, \infty\}$ for a portal p and two neighboring portals differ by at most l/m. We assign ∞ as a value for $\text{inside}_i(p)$, if no node of level i is inside the corresponding box. Hence, we have at most $2m \cdot 3^{4m}$ possible assignments per box for each inside_i function.

A slightly different reasoning holds for the outside_i functions. Here, the maximal distance from a portal to an outside node can be at most $2L$. Again, we can allow an additional additive error of l/m. Hence, we have $\text{outside}_i(p) \in$

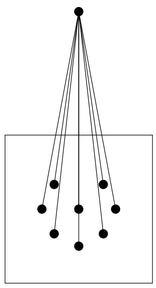

Figure 4.3: All nodes actually lie on top of each other and the edges pass through one portal.

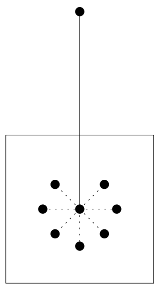

Figure 4.4: All nodes actually lie on top of each other and the dotted lines have length 0.

$\{0, l/m, 2l/m, \ldots, 2L, \infty\}$. This sums up to at most $2Lm/l$ different values and at most $2Lm/l \cdot 3^{4m}$ possible assignments per box for each outside_i function. This could be reduced by making the gap between two consecutive values larger as the distance becomes larger, since for larger distances we anyway have a larger additional error due to a larger interportal distance, but we omit this here. In total we have $T = 4Lm^2 \cdot 3^{8mk}$ entries in table Table per box B.

Computing the table

We compute the table Table bottom up. There are two different base cases:

1. The root r is inside the box B.

We set $\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$ to cost 0 if the following two conditions both hold

1. $\text{inside}_0(p)$ is the distance from each portal p to the root r, and
2. $\text{inside}_i(p)$ for $i \geq 1$ is ∞ for all p.

Should at least one of the above conditions not hold, we set its corresponding Table entry to ∞.

2. The box B contains at least one node but no root.

Note that all nodes lie in the center of box B and thus on top of each other due to the initial perturbation. Let one of these nodes be node q. If $\text{inside}_i(p)$ is the distance between the nodes in the box and each portal p for all p and some $i \geq 1$, and $\text{inside}_{i'}(p) = \infty$ for all $i' \neq i$, and all portals p, we then connect node q to the portal p' such that $\text{dist}(q, p') + \text{outside}_{i-1}(p')$ is minimal among all portals of this box. We store this cost in the corresponding Table entry. If however, all $\text{inside}_i(p)$ are ∞ for all i and all p, i.e. this box does not provide any reachable node to the outside, we have to distinguish two cases. In the first case it is cheaper to connect all nodes to a level $k-1$ node as depicted in Figure 4.3. In the second case, it is cheaper to connect one node q to a node of level at most $k-2$ and then connecting all other nodes inside the box to this node q as in Figure 4.4. Which case we have can be determined by looking at the corresponding outside_i functions. We store the cost in the corresponding Table entry. In all other cases, i.e. the inside_i and outside_i functions do not fully satisfy one of the above cases, we set the corresponding Table entry to ∞, since such a configuration can never be satisfied.

If we are not in the base case, the entry of $\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$ can be computed from the corresponding table entries $\text{Table}(B_j, \text{inside}_0^{(j)}, \ldots, \text{inside}_{k-1}^{(j)} \text{outside}_0^{(j)}, \ldots, \text{outside}_{k-1}^{(j)})$, for $1 \leq j \leq 4$, where B_1, B_2, B_3 and B_4 are the four sub-boxes of B, and $\text{inside}_i^{(j)}$ and $\text{outside}_i^{(j)}$ are the corresponding inside and outside functions of level i of sub-box B_j. Once all inside_i and outside_i functions are fixed we go through all possible $\text{inside}_i^{(j)}$ and $\text{outside}_i^{(j)}$ functions that comply with distance functions of box B. As we only have approximate distances stored we again introduce an additive error of at most l/m per line segment. However, this error is at most the error that occurs while making an edge portal-respecting for this box and hence, can be neglected here. We sum up the corresponding costs for B_1, B_2, B_3 and B_4 and store the minimal in the corresponding $\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$ entry. The time spent per box amounts then to $O(T^5)$.

As there are L^2 boxes in the dissection the total running time amounts to $O(L^2 \cdot T^5) = (\frac{n}{\epsilon})^{O(k/\epsilon)}$.

We conclude with the main theorem:

Theorem 4.3 *The k-hop minimum spanning tree problem in the Euclidean plane admits a polynomial time approximation scheme for any fixed k.*

4.4 Generalizations

The bounded-depth minimum Steiner tree problem

Our approach easily generalizes to the shallow Steiner tree problem. Here, one is also allowed to use Steiner points in the bounded-hop MST. We just have to change the base case in our algorithm. If we only have Steiner points inside a box we have two options: either use the Steiner point or do not use it. This can be easily decided based on the distance functions on the portals.

The multi-level concentrator location problem

If we assign levels to the Steiner points and also opening costs for using a Steiner point, we are left with the multi-level concentrator location problem. This problem can also be solved using our approach. We just have to add the opening cost to the corresponding Table entry. For the initial perturbation it suffices to have a lower bound on the optimal cost which is polynomial in the number of nodes n. The corresponding k-level facility location problem obviously is an n-approximation. Aardal et al. [ACS99] showed how to compute a 3-approximation for this problem. Hence, the initial bounding box has size $L = 3n^2/\epsilon$, and the running time adapts accordingly.

4.5 Conclusions and Open Problems

We provided the first polynomial time approximation scheme for the k-hop minimum spanning tree and related problems in the plane. The algorithm follows along the lines of Arora et al. [ARR98]. Thus, the algorithm can be generalized to higher dimensions but with only quasi-polynomial running time. It would be interesting to find a PTAS also for higher dimensions. As mentioned before, Kolliopoulos and Rao in [KR07] construct a randomized PTAS for k-median problem in the d-dimensional Euclidean space. However, their Structure theorem and algorithm is based on an adaptive dissection that guesses at every level the solution to the optimal facility assignment. Unfortunately, it is not obvious how to adapt their dissection to problems like the k-hop MST problem.

Bibliography

[AAB$^+$06] Helmut Alt, Esther M. Arkin, Herve; Brönnimann, Jeff Erickson, Sandor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. Minimum-cost coverage of point sets by disks. In *SCG '06: Proc. 22nd Ann. Symp. on Computational Geometry*, pages 449–458, New York, NY, USA, 2006. ACM Press.

[ACI$^+$04] Christoph Ambühl, Andrea E. F. Clementi, Miriam Di Ianni, Nissan Lev-Tov, Angelo Monti, David Peleg, Gianluca Rossi, and Riccardo Silvestri. Efficient algorithms for low-energy bounded-hop broadcast in ad-hoc wireless networks. In *STACS*, pages 418–427, 2004.

[ACS99] Karen Aardal, Fabian A. Chudak, and David B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Inf. Process. Lett.*, 72(5-6):161–167, 1999.

[AFHP$^+$05] Ernst Althaus, Stefan Funke, Sariel Har-Peled, Jochen Könemann, Edgar A. Ramos, and Martin Skutella. Approximating k-hop minimum-spanning trees. *Operations Research Letters*, 33(2):115–120, March 2005.

[Amb05] Christoph Ambühl. An optimal bound for the mst algorithm to compute energy-efficient broadcast trees in wireless networks. In *ICALP*, pages 1139–1150, 2005.

[And01] Thomas Andreae. On the traveling salesman problem restricted to inputs satisfying a relaxed triangle inequality. *Networks*, 38(2):59–67, 2001.

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[ARR98] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 106–113, New York, NY, USA, 1998. ACM Press.

[Ass83] P. Assouad. Densité et dimension. *Ann. Inst. Fourier*, 33(3):233–282, 1983.

[BC00] Michael A. Bender and Chandra Chekuri. Performance guarantees for the TSP with a parameterized triangle inequality. *Information Processing Letters*, 73(1–2):17–21, 2000.

[BCKK05] Vittorio Bilò, Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Geometric clustering to minimize the sum of cluster sizes. In *ESA*, pages 460–471, 2005.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[BG94] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension: (preliminary version). In *SoCG '94*, pages 293–302, New York, NY, USA, 1994. ACM Press.

[CCP+01] Andrea E. F. Clementi, Pierluigi Crescenzi, Paolo Penna, Gianluca Rossi, and Paola Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In *STACS*, pages 121–131, 2001.

[CGMZ05] Hubert T.-H. Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. In *SODA*, pages 762–771, 2005.

[CHP+02] A. Clementi, G. Huiban, P. Penna, G. Rossi, and Y. Verhoeven. Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In *Proc. 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE), pages 23-38.*, 2002.

[CHR+03] Andrea E. F. Clementi, Gurvan Huiban, Gianluca Rossi, Yann C. Verhoeven, and Paolo Penna. On the approximation ratio of the mst-based heuristic for the energy-efficient broadcast problem in static ad-hoc radio networks. In *IPDPS*, page 222, 2003.

[CIM+05] Andrea E. F. Clementi, Miriam Di Ianni, Angelo Monti, Massimo Lauria, Gianluca Rossi, and Riccardo Silvestri. Divide and conquer is almost optimal for the bounded-hop mst problem on random euclidean instances. In Andrzej Pelc and Michel Raynal, editors, *SIROCCO*, volume 3499 of *Lecture Notes in Computer Science*, pages 89–98. Springer, 2005.

[CK95] Paul B. Callahan and S. Rao Kosaraju. Algorithms for dynamic closest pair and n-body potential fields. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 1995.

[Cla95] K. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995.

[CV05] K. L. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. In *SoCG '05*, pages 135–141, New York, NY, USA, 2005. ACM Press.

[DC90] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Trans. Comput. Syst.*, 8(2):85–110, 1990.

[DEF+94] Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. An architecture for wide-area multicast routing. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 126–135, New York, NY, USA, 1994. ACM Press.

[EJS01] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric graphs. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pages 671–679, Washington, DC, 7–9 2001.

[ERS05] G. Even, D. Rawitz, and S. Shahar. Hitting sets when the vc-dimension is small. *Inf. Process. Lett.*, 95(2):358–362, 2005.

[EW85] H. Edelsbrunner and E. Welzl. On the number of line separations of a finite set in the plane. *J. Comb. Theory, Ser. A*, 38(1):15–29, 1985.

[Fei98] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[FG88] T. Feder and D. Greene. Optimal algorithms for approximate clustering. *Proc. 20th ACM Symp. on Theory of Computing*, 1988.

[FGNW06] S. Funke, L. J. Guibas, A. Nguyen, and Y. Wang. Distance-sensitive information brokerage in sensor networks. In *to appear in Proc. IEEE International Conference on Distributed Computing in Sensor System (DCOSS'06)*, 2006.

[FL07] Stefan Funke and Sören Laue. Bounded-hop energy-efficient broadcast in low-dimensional metrics via coresets. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, pages 272–283, 2007.

[FLN07] Stefan Funke, Sören Laue, and Rouven Naujoks. Minimum-energy broadcast with few senders. In James Aspnes, Christian Scheideler, Anish Arora, and Samuel Madden, editors, *DCOSS*, volume 4549 of *Lecture Notes in Computer Science*, pages 404–416. Springer, 2007.

[FLNL08] Stefan Funke, Sören Laue, Rouven Naujoks, and Zvi Lotker. Power assignment problems in wireless communication: Covering points by disks, reaching few receivers quickly, and energy-efficient travelling salesman tours. In *DCOSS*, volume 5067 of *Lecture Notes in Computer Science*. Springer, 2008.

[FMS03] Stefan Funke, Domagoj Matijevic, and Peter Sanders. Approximating energy efficient paths in wireless multi-hop networks. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003: 11th Annual European Symposium*, volume 2832 of *Lecture Notes in Computer Science*, pages 230–241, Budapest, Hungary, September 2003. Springer.

[FNKP04] Michele Flammini, Alfredo Navarra, Ralf Klasing, and Stéphane Pérennes. Improved approximation results for the minimum energy broadcasting problem. In *DIALM-POMC*, pages 85–91, 2004.

[FRT03] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455, New York, NY, USA, 2003. ACM Press.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, January 1979.

[GK98] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *FOCS '98*, page 300, Washington, DC, USA, 1998. IEEE Computer Society.

[GK99] Sudipto Guha and Samir Khuller. Greedy strikes back: improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.

[GKK+08] Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi Varadarajan. On clustering to minimize the sum of radii. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 819–825, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[GKL03] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 534, Washington, DC, USA, 2003. IEEE Computer Society.

[Har04] S. Har-Peled. Clustering motion. *Discrete Comput. Geom.*, 31(4):545–565, 2004.

[HM85] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *J. ACM*, 32(1):130–136, 1985.

[HM04] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *STOC*, pages 291–300, 2004.

[HM06] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

[KKKP00] Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. Power consumption in packet radio networks. *Theor. Comput. Sci.*, 243(1-2):289–305, 2000.

[KMY03] Piyush Kumar, Joseph S. B. Mitchell, and E. Alper Yildirim. Approximate minimum enclosing balls in high dimensions using coresets. *J. Exp. Algorithmics*, 8:1.1, 2003.

[KP06] Erez Kantor and David Peleg. Approximate hierarchical facility location and applications to the shallow steiner tree and range assignment problems. In Tiziana Calamoneri, Irene Finocchi, and Giuseppe F. Italiano, editors, *CIAC*, volume 3998 of *Lecture Notes in Computer Science*, pages 211–222. Springer, 2006.

[KPP93] Vachaspathi P. Kompella, Joseph C. Pasquale, and George C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Netw.*, 1(3):286–292, 1993.

[KPW92] J. Komlós, J. Pach, and G. J. Woeginger. Almost tight bounds for epsilon-nets. *Discrete and Computational Geometry*, 7:163–173, 1992.

[KR99] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean kappa-median problem. In *ESA '99: Proceedings of the 7th Annual European Symposium on Algorithms*, pages 378–389, London, UK, 1999. Springer-Verlag.

[KR07] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. *SIAM J. Computing*, 37:757–782, 2007.

[Lau08] Sören Laue. Geometric set cover and hitting sets for polytopes in r^3. In Susanne Albers, Pascal Weil, and Christine Rochange, editors, *STACS*, volume 08001 of *Dagstuhl Seminar Proceedings*, pages 479–490. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.

[LM08] Sören Laue and Domagoj Matijević. Approximating k-hop minimum spanning trees in euclidean metrics. *Information Processing Letters*, 2008. to appear.

[Mat92] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2(3):169–186, 1992.

[Mat02] J. Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[MSW90] J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: Small epsilon-nets for disks and halfspaces. In *SoCG '90*, pages 16–22, 1990.

[MYZ02] Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. Improved approximation algorithms for metric facility location problems. In *APPROX '02: Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 229–242, London, UK, 2002. Springer-Verlag.

[OM] P. Orponen and H. Manila. On approximation preserving reductions: Complete problems and robust measures. Technical Report, University of Helsiki, 1990.

[PA95] J. Pach and P. K. Agarwal. *Combinatorial Geometry*. Wiley, New York, 1995.

[Pat00] D. Patel. Energy in ad-hoc networking for the picoradio. Master's thesis, UC Berkeley, 2000.

[PW90] J. Pach and G. Woeginger. Some new bounds for epsilon-nets. In *SoCG '90*, pages 10–15, New York, USA, 1990. ACM Press.

[Rap96] T.S. Rappaport. *Wireless Communication*. Prentice Hall, 1996.

[SK99] S.Guha and S. Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and Computation*, 150:57–74, 1999.

[Tal04] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *STOC*, pages 281–290, 2004.

[VC71] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probability. *Theory Probab. Appl.*, 16:264–280, 1971.

[WCLF01] Peng-Jun Wan, Gruia Calinescu, Xiangyang Li, and Ophir Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *INFOCOM*, pages 1162–1171, 2001.

[WNE00] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM*, pages 585–594, 2000.

[You95] N. E. Young. Randomized rounding without solving the linear program. In *SODA '95*, pages 170–178, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[Zha04] Jiawei Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 808–817, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.