

Covering Problems via Structural Approaches

by

Elyot Grant

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2011

©Elyot Grant 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The minimum set cover problem is, without question, among the most ubiquitous and well-studied problems in computer science. Its theoretical hardness has been fully characterized—logarithmic approximability has been established, and no sublogarithmic approximation exists unless $\mathbf{P} = \mathbf{NP}$. However, the gap between real-world instances and the theoretical worst case is often immense—many covering problems of practical relevance admit much better approximations, or even solvability in polynomial time. Simple combinatorial or geometric structure can often be exploited to obtain improved algorithms on a problem-by-problem basis, but there is no general method of determining the extent to which this is possible.

In this thesis, we aim to shed light on the relationship between the structure and the hardness of covering problems. We discuss several measures of structural complexity of set cover instances and prove new algorithmic and hardness results linking the approximability of a set cover problem to its underlying structure. In particular, we provide:

- An **APX**-hardness proof for a wide family of problems that encode a simple covering problem known as **Special-3SC**.
- A class of polynomial dynamic programming algorithms for a group of weighted geometric set cover problems having simple structure.
- A simplified *quasi-uniform sampling algorithm* that yields improved approximations for weighted covering problems having low *cell complexity* or *geometric union complexity*.
- Applications of the above to various *capacitated covering problems* via linear programming strengthening and rounding.

In total, we obtain new results for dozens of covering problems exhibiting geometric or combinatorial structure. We tabulate these problems and classify them according to their approximability.

Acknowledgements

I wish foremost to thank my supervisors Jochen Koenemann and Timothy Chan for providing guidance and sharing their knowledge of combinatorial optimization and computational geometry with me. Without the blending of wisdom from these two areas, much of this work would not have been possible. I additionally would like to acknowledge my thesis readers Therese Biedl and Chaitanya Swamy, my collaborators Deeparnab Chakrabarty and Malcolm Sharpe, and the many anonymous referees whose comments have helped improve both the strength and the presentation of the results herein. Finally, I wish to thank (in no particular order) Adrian Bock, Laura Sanità, Nick Harvey, Will Ma, David Rhee, Shalev Ben-David, Alex Wice, Abel Molina, Alyssa Carey, David Pritchard, and countless other supporting friends, colleagues, and family members who have been sources of rich mathematical discussions, insightful advice, and much-needed motivation.

I graciously acknowledge the Natural Sciences and Engineering Research Council of Canada for partially supporting this work.

Contents

1	Introduction	1
1.1	Set Cover: A Hard Problem with a Rich History	2
1.2	Motivating Problems and Applications	7
1.2.1	Interval Cover	8
1.2.2	Rectangle Cover	10
1.2.3	Tree Cover	14
1.3	Organization of Thesis	16
2	Foundations	19
2.1	Set System Matrices	19
2.1.1	Duality	20
2.1.2	Heredity	20
2.1.3	Subproblems	21
2.1.4	The Integrality Gap	22
2.1.5	Total Unimodularity	23
2.2	Complexity-Theoretic Preliminaries	25
2.3	Set Systems and their Structural Properties	29
2.3.1	Bounded Degree and Frequency	29
2.3.2	Hitting Set and Dominating Set	30
2.3.3	Combining and Disassembling Set Systems	32
2.3.4	Geometric Set Systems	35
2.3.5	Priority Set Systems	40
2.4	Measuring and Exploiting the Complexity of Set Systems	42
2.4.1	VC dimension	43
2.4.2	ϵ -Nets and the Theorem of Haussler and Welzl	47
2.4.3	From ϵ -Nets to Set Covers	49
3	Related Work	53
3.1	Known Hardness Results	53
3.2	ϵ -Net Finders for Geometric Set Cover	55
3.3	The Pach-Tardos Counterexample	59
3.4	The Local Search Approach	61
3.5	Other Methods for Geometric Set Cover	62

4	Polynomial-Time Algorithms for Geometric Set Cover	65
4.1	Main Dynamic Programming Algorithm	65
4.2	Applications: Half-planes, Pseudodisks, and More	68
4.3	Algorithm for a Hitting Set Problem	70
5	APX-Hardness Results	71
5.1	Special-3SC	71
5.2	Encodings of Special-3SC via Geometric Set Cover	73
5.3	Hardness of Priority Tree Cover	77
6	Improved Quasi-Uniform Sampling for Weighted Set Cover	79
6.1	Shallow Cell Complexity	79
6.2	The Quasi-Uniform Sampling Algorithm	80
6.3	Derandomization	87
6.4	Geometric Set Systems with Low Shallow Cell Complexity	88
6.5	Shallow Cell Complexity of Other Set Systems	89
7	Capacitated and Priority Covering Problems	93
7.1	Preliminaries and Statement of Main Result	94
7.2	Proof of Main Result	97
7.2.1	Strengthening the Canonical LP Relaxation	98
7.2.2	Solving the Residual Problem	99
7.2.3	Dealing with Small Rows via Multi-Cover	100
7.2.4	Dealing with Large Rows via Priority Covering	104
7.2.5	Putting Everything Together	104
7.3	Applications	105
7.4	Combinatorial Approximation Algorithm for Priority Tree Cover	106
8	Summary and Conclusion	109
8.1	Summary of Results	110
8.2	Directions for Future Research	112
	Bibliography	115

Chapter 1

Introduction

Many computationally intractable problems in discrete optimization suffer from a large gap between their best known theoretical approximability and the experimental results obtained by practical approaches applied to real-world instances. Consider, as a familiar example, the *Travelling Salesman Problem (TSP)*—arguably one of the most widely known and well-studied problems in theoretical computer science. The standard (metric, symmetric, weighted) version of the TSP admits a 1.5-approximation via the famous algorithm of Christofides, but an improved approximation remains unknown in general [Vaz01]. However, well-optimized TSP solvers designed for practical purposes can often, in a short amount of time, obtain results that are provably within a fraction of a percent of optimal on instances containing hundreds of thousand of vertices [JM02]. Randomly generated or real-world TSP instances are typically quite tractable from an approximability point of view.

This theme is repeated, perhaps in an even more dramatic fashion, for *covering problems* such as *Minimum Set Cover* and *Minimum Hitting Set*—the main topics of this thesis. As we shall see, in the general case, such problems are inapproximable within anything better than a multiplicative logarithmic factor unless $\mathbf{P} = \mathbf{NP}$. However, there are numerous special cases of practical relevance in which geometric or combinatorial constraints are present, and consequently, much better approximations, or even exact solution in polynomial time, are possible. Moreover, even for cases where theoretically good approximations are not known, there often exist heuristic algorithms that achieve astonishingly good results on instances generated randomly or obtained from industrial applications; a compelling example can be found in [HLL06] for *Rectilinear Polygon Cover*—a problem discussed at some length in Section 1.2.

These examples beg several fundamental questions that are of immense importance in operations research and computer science. Why are typical or randomly generated instances of computational optimization problems often significantly easier than the theoretical worst case? What aspect of pathological cases makes them so difficult, and why do such difficulties seem to occur rarely in practice? What must we understand in order to establish theoretical jus-

1. INTRODUCTION

tification for the easiness of practical problems in the face of strong hardness results?

For the case of covering problems, we shall argue that the answer to all of the above questions can be distilled down to a single word: *structure*. Set cover problems encountered in practical applications typically have structural restrictions that forbid the existence of overly complex subproblems. Consequently, pathological cases fail to present themselves, and instead, instance-specific properties can be exploited to obtain better approximation algorithms.

Indeed, many such improved approximation algorithms exist for special cases of set cover arising in real-world geometric or combinatorial optimization problems. However, such structure-exploiting methods are frequently obtained in an ad hoc manner on a problem-by-problem basis. There is no general method of detecting underlying structure present within set cover instances and capitalizing on it for the purposes of obtaining the best possible approximability. Furthermore, the current state of the art possesses few reliable and general methods of easily determining whether or not a given covering problem admits any improved approximability at all.

In this thesis, we aim to present simpler and more general tools for obtaining both upper and lower bounds on the approximability of various classes of set cover problems. Our results are achieved by establishing connections between the underlying combinatorial structure of instances and their approximability (or hardness of approximation). Accordingly, we obtain new polynomial-time solvability and **APX**-hardness results for wide classes of covering problems as well as improved approximation algorithms whose guarantees depend on various instance-specific parameters.¹ In doing so, we acquire dozens of improved results for numerous covering problems in computational geometry and combinatorial optimization, including several weighted and capacitated covering problems.

1.1 Set Cover: A Hard Problem with a Rich History

We begin by reviewing the formal definition of minimum set cover and the related family of optimization problems. We start with the basics:

Definition 1.1.1. A *set system* is an ordered pair (X, \mathcal{S}) where X is any set and \mathcal{S} is a family of subsets of X .

Topologies, graphs, matroids, and so on can be regarded as special cases of set systems, but for the purposes of defining general set cover, we shall impose no structural restrictions of any kind on \mathcal{S} . Set systems are also known as *hypergraphs* or *range spaces*, with the distinction depending only on context

¹Such *parameterized approximation algorithms*, as they are called, are well studied (see, e.g. [CGG06]) However, we will not dwell on the theoretical aspects of parameterized approximability, instead choosing to focus on what these methods can do for various individual covering problems.

(combinatorics versus geometry). The set X , often called the *universe* or *ground set*, will be assumed finite for all of our applications. Accordingly, \mathcal{S} will always be a finite family containing finite sets. In the literature, members of X are usually called *points*, *elements*, or *atoms*; members of \mathcal{S} are commonly called *ranges* or simply *sets*. In the terminology of hypergraphs, members of X may be called *vertices* or *nodes* and members of \mathcal{S} may be called *hyperedges* (or sometimes just *edges*). Throughout this thesis, we will use the variable M for $|X|$ —the number of elements—and the variable N for $|\mathcal{S}|$ —the number of sets.²

When defining set systems arising in geometric applications, we will employ a slight form of notational abuse. In such cases, sets in \mathcal{S} will not be defined explicitly, but will be inferred from other classes of geometric objects or regions. For example, X may be a finite set of points in the plane, and \mathcal{S} could be regarded as a family of disks, squares, or other objects, each of which contains some points in X . When discussing such set systems, we will treat the sets in \mathcal{S} as geometric objects (and even refer to them as *objects*, *regions*, *disks*, and so on). However, on a purely theoretical level, we shall only be concerned with the restriction of each set in \mathcal{S} to X .

In the optimization problem **Min-Set-Cover**, we are given a set system (X, \mathcal{S}) and must select a subfamily $\mathcal{C} \subseteq \mathcal{S}$ such that each element in X lies inside at least one set in \mathcal{C} . The objective that we wish to minimize is the cardinality of \mathcal{C} . In the related **Min-Hitting-Set** problem, we instead wish to select a minimum cardinality subset $Y \subseteq X$ such that each set in \mathcal{S} contains at least one point in Y . For both of these problems to always have a feasible solution, we shall assume throughout this thesis that each element in X is contained in at least one set in \mathcal{S} , and that no sets in \mathcal{S} are empty.

In the literature, **Min-Set-Cover** and **Min-Hitting-Set** are sometimes referred to as *minimum hypergraph cover* or *minimum hypergraph transversal* respectively. We will not use these terms.

In the *weighted* generalizations of **Min-Set-Cover** and **Min-Hitting-Set**, we are also given a vector of positive *costs* or *weights* $\mathbf{w} \in \mathbb{R}_+^{\mathcal{S}}$ or $\mathbf{w} \in \mathbb{R}_+^X$ and we wish to minimize the total cost of all objects in \mathcal{C} or Y respectively. Instances without costs (or, equivalently, with unit costs) are termed *unweighted*. In this thesis, we will assume a model of computation in which all weights admit a representation allowing primitive operations (addition and comparison) in time linear in their size. If desired, the reader may simply assume that all weights are positive integers. Wherever possible, we shall state hardness results for unweighted versions of problems and algorithmic results for weighted versions so as to obtain the strongest possible theorems. When simply developing structural properties of set systems, the distinction will remain unimportant, so we will only distinguish between weighted and unweighted problems where necessary.

It turns out that the two problems **Min-Set-Cover** and **Min-Hitting-Set** are equivalent under the interchanging of the roles of X and \mathcal{S} . Formally:

²This is in agreement with notation used in some publications and directly opposite to that used in others (there appears to be no clear favourite in the literature). We choose to make M the number of points so that when we introduce *set system matrices* for the integer programming formulation of set cover, they will be M by N matrices.

1. INTRODUCTION

Definition 1.1.2. Given a set system (X, \mathcal{S}) , its *dual set system*³ $(X, \mathcal{S})^*$ is the set system (Z, \mathcal{T}) with $Z = \{z_S : S \in \mathcal{S}\}$ and $\mathcal{T} = \{T_x : x \in X\}$ where $T_x = \{z_S : x \in S\}$.

One way to visualize the dual set system is as follows: first, regard a set system (X, \mathcal{S}) as a bipartite graph with vertex set $A \cup B$, where the elements of A are members of X , elements of B are members of \mathcal{S} , and edges in the graph correspond to the inclusion of elements in sets. A set cover then corresponds to a family of vertices in B whose neighbours include the entirety of A , whereas a hitting set corresponds to a family of vertices in A whose neighbours include the entirety of B . The dual set system is then the set system obtained by interchanging the roles of A and B . It should thus be clear that $((X, \mathcal{S})^*)^* = (X, \mathcal{S})$. Moreover, the following is immediately apparent:

Proposition 1.1.3. *The problem **Min-Set-Cover** on the set system (X, \mathcal{S}) is isomorphic to the problem **Min-Hitting-Set** on the set system $(X, \mathcal{S})^*$.*

In the above, by isomorphic, we mean that there is a structure-preserving bijection between the two instances that preserves the feasibility and objective value of potential solutions.

Remark 1.1.4. Proposition 1.1.3 even holds for the weighted generalizations of **Min-Set-Cover** and **Min-Hitting-Set** as long as the same weights are used for both problems.

Proposition 1.1.3 indicates that, from a computational point of view, **Min-Set-Cover** and **Min-Hitting-Set** are essentially the same problem. For clarity and consistency in our exposition, we shall focus on **Min-Set-Cover** when discussing theoretical results, with the implicit knowledge that our theorems are applicable to appropriate hitting set problems after taking the dual set system. However, when discussing specific classes of set systems—particularly those obtained from geometric situations—we will often explicitly mention hitting set formulations in cases where the dual set system lacks an intuitive geometric formulation. We shall informally use the umbrella term *covering problem* (or sometimes, the more specific expressions *weighted covering problem* and *unweighted covering problem*) for any computational optimization problem with an obvious, direct formulation as a **Min-Set-Cover** or **Min-Hitting-Set** problem. Such problems include many classical graph-theoretic optimization problems like minimum vertex cover, minimum edge cover, minimum dominating set, minimum clique cover, and so on.

Covering problems also have a natural formulation as an *integer program*. The most obvious way of defining set systems in this context is via binary matrices that encode the element-set incidence relation:

³A cautionary note: the term ‘dual’ in the definition of dual set system has nothing to do with linear programming duality. A better term (given the integer programming formulation of set cover) would be ‘transpose’, but we stick with the classical terminology.

Definition 1.1.5. For a set system (X, \mathcal{S}) with $X = \{v_1, \dots, v_M\}$ and $\mathcal{S} = \{S_1, \dots, S_N\}$, we define the *set system matrix*⁴ $A_{(X, \mathcal{S})}$ to be the M by N matrix whose entry in position (i, j) is 1 if element v_i is contained in set S_j and 0 otherwise.

As mathematical objects, we will always consider set system matrices *modulo row and column reorderings* so as to ensure they are well defined and to establish a bijective correspondence between set system isomorphism classes and set system matrices. For our purposes, it may be assumed that set system matrices contain no duplicate rows or columns.

If we wish to obtain an optimal solution to the weighted **Min-Set-Cover** problem on a set system having matrix A and weights \mathbf{w} , then it suffices to solve the following integer program:

$$\min\{\mathbf{w}^T \mathbf{x} : A\mathbf{x} \geq \mathbf{1}, \mathbf{x} \in \{0, 1\}\} \quad (\text{SCIP})$$

The following linear programming relaxation of (SCIP) can be solved in polynomial time to produce a minimum cost *fractional set cover*:

$$\min\{\mathbf{w}^T \mathbf{x} : A\mathbf{x} \geq \mathbf{1}, \mathbf{x} \geq \mathbf{0}\} \quad (\text{SCLP})$$

It is clear that the optimal objective value of (SCLP) is a lower bound on the optimal objective value of (SCIP). Properties of (SCIP) and its linear programming relaxation will be crucial when establishing the structural properties of various covering problems. Section 2.1 provides an overview of the important facts.

More general integer programs similar to (SCIP) may be obtained by replacing A with an arbitrary (not necessarily binary) matrix and replacing $\mathbf{1}$ with an arbitrary right hand side vector. Such problems are usually called *covering integer programs* [Vaz01] or CIPs. Chapter 7 contains some results for CIPs more general than set cover; for the remainder of this thesis, we shall deal only with covering problems in which the matrices and right hand sides in (SCIP) are binary.

Covering problems have a rich history of intractability spanning multiple decades. Appropriate decision versions of the unweighted **Min-Set-Cover** and **Min-Hitting-Set** problems were first proven to be **NP**-complete in 1972 by Karp in his famous article in which **NP**-completeness proofs for 21 combinatorial problems were given [Kar72]. With the development of the PCP theorem and related inapproximability theory in the 1990s, many stronger hardness results for set cover soon followed. Papadimitriou and Yannakakis proved in 1988 that **Min-Set-Cover** is hard for the complexity class **MAX-SNP** [PY91]; together with results from a 1992 paper of Arora et al. [ALM⁺98], this showed that the problem admits no polynomial time approximation scheme (PTAS) unless $\mathbf{P} = \mathbf{NP}$. In 1994, Lund and Yannakakis provided the first superconstant

⁴In the combinatorics literature, such objects are sometimes simply called *incidence matrices*; we refrain from using this terminology to avoid confusion with vertex-edge incidence matrices of graphs.

1. INTRODUCTION

lower bound on the approximability of set cover by showing that no polynomial time algorithm can achieve an approximation factor better than $\frac{1}{2} \log_2 M$, or approximately $0.72 \ln M$ (where M is the number of elements in the set cover instance) [LY94]. However, their result requires the assumption that \mathbf{NP} has no randomized quasi-polynomial time algorithms.⁵ Feige subsequently improved upon the construction of Lund and Yannakakis to show an inapproximability threshold of $(1 - o(1)) \ln M$ under an identical complexity assumption [Fei98]. Around the same time, Raz and Safra provided a weaker lower bound of $c \cdot \ln n$ for a small positive value of c under the weaker complexity-theoretic assumption that $\mathbf{P} \neq \mathbf{NP}$ [RS97]. Using more recent techniques, this result was reproven for a larger value of c by Alon, Moshkovitz, and Safra [AMS06].

As it turns out, classical approximation algorithms for set cover produce results that agree almost exactly with these inapproximability bounds. Independently in the 1970s, Johnson [Joh74] and Lovász [Lov75] both proved that the simple heuristic of repeatedly choosing the set containing the most uncovered elements produces a $(\ln M + 1)$ -approximation for unweighted **Min-Set-Cover**. A simple generalization of this to the weighted case by Chvátal in 1979 [Chv79] gave the same approximation threshold. In fact, logarithmic approximations for weighted set cover can be achieved using almost every standard approximation technique, including randomized methods, iterated linear programming rounding, primal-dual schema, and so on (see [Vaz01] for a thorough listing). Moreover, all of the aforementioned techniques, including that of the simple greedy algorithm, can be shown (e.g. by the method of dual fitting) to give approximability results relative to the optimal value of the linear programming relaxation (SCLP).

The performance of these algorithms seems to indicate that there is little remaining to learn about the general set cover problem from an approximability point of view, aside from resolving the complexity-theoretic hypotheses required for the inapproximability proofs to go through. In particular, the $(1 - o(1)) \ln M$ -factor inapproximability result of Feige is tight up to lower order terms, barring the discovery of a pseudopolynomial algorithm for \mathbf{NP} -complete problems, which appears unlikely given current state of the art (though it certainly has not been ruled out). Accordingly, this same assumption implies the surprising and somewhat depressing conclusion that no polynomial algorithm can ever outperform the trivial greedy method on general instances by anything more than a subconstant factor.

Fortunately, many improvements to the greedy method are possible when covering problems become more structured. From a practical point of view, the theoretical inapproximability of the general **Min-Set-Cover** problem often poses few obstacles in obtaining good approximations for real-world instances. Additionally, many covering formulations of other optimization problems have good approximability properties. Throughout this thesis, we will explore numerous covering problems with structural restrictions that enable us to obtain better approximations. We first motivate our work with some simple examples.

⁵Specifically, the result holds unless $\mathbf{NP} \subseteq \mathbf{ZTIME}(n^{\text{polylog}(n)})$.

1.2 Motivating Problems and Applications

Covering problems are ubiquitous in computer science and optimization because they conveniently model any situation in which a variety of independent demands must be met by selecting some subset of a list of available options. For example, an immediate and obvious application of geometric covering is to wireless network planning, where one might be asked to cheaply position a configuration of cellular antennas or wireless routers to provide service to clients [GRV05]. We can model this by representing each client via an element, and constructing a set for each feasible location in which an antenna could be installed (containing the elements corresponding to the clients that it could serve). If we wish to allow for the option of paying a fee to avoid servicing a client, we can simply add singleton sets with appropriate fees. Depending on the physical properties of the antennas, we could make assumptions about geometric structure and use them to obtain an improved approximation. For example, we shall see in Chapter 6 that if the area of coverage of each antenna can be assumed to be a disk in the plane, then the corresponding covering problem admits a constant approximation.

Of course, **Min-Set-Cover** is **NP**-complete and so can theoretically encode any problem in **NP** via a sufficiently long chain of reductions. However, we avoid dealing directly with set cover formulations of problems that are not, by their very nature, covering problems. We also deliberately overlook many covering formulations of optimization problems where the number of variables or sets becomes exponential in the original problem size. For example, a common technique in many *network design* problems is to express connectivity requirements on a graph G as covering-type requirements over the exponential-sized set of cuts in G , and then obtain LP-based approximations.⁶ We will not study such techniques, or the covering problems these formulations generate. In this thesis, we assume all set systems are represented explicitly in the input.

To clarify our intentions, we shall now provide a few representative problems that exemplify the types of structure we will be studying. We examine them for many reasons: first, to familiarize the reader with the type of problems we shall study and the conventions we shall use; secondly, to introduce and explain several simplifying assumptions that will be applied repeatedly throughout this thesis; and finally, to provide background motivation for studying such problems in the first place, by giving applications of both theoretical and real-world importance. It shall become clear that these types of covering problems conceal a variety of fascinating connections among algorithms, complexity theory, geometry, combinatorics, and even topology.

⁶Despite its exponential size in these cases, the linear programming relaxation (SCLP) can often be solved in polynomial time. This requires the use of the ellipsoid method, which can, in certain cases, produce an optimal basic feasible solution in polynomial time as long as polynomial separation oracles exist. See [Vaz01] for more information and many examples of this method in action.

1.2.1 Interval Cover

The first problem we discuss is a trivial polynomial-time solvable covering problem that we shall refer to as **\mathbb{R} -Interval-SC**:

COVERING PROBLEM: **\mathbb{R} -Interval-SC**
 ELEMENTS: A finite subset of \mathbb{R}
 SETS: A family of intervals in \mathbb{R}

The above formatting shall serve as the standard manner in which we introduce and describe different classes of set systems, and accordingly, different covering problems. To define the class of set systems, we provided a simple (in this case, geometric) embedding of X into a larger set (\mathbb{R}), and then asserted restrictions on the structure of the family of sets \mathcal{S} (they must be a family of *intervals*). The suffix ‘**SC**’ that follows then indicates that we are concerned with the **Min-Set-Cover** problem on this class of set systems. We will employ other suffixes for hitting set and dominating set problems (see Section 2.3.2).

The **\mathbb{R} -Interval-SC** problem itself is usually referred to as *line cover* or *interval cover*. We highlight several important points:

- We do not define the sets \mathcal{S} explicitly as a family of subsets of the universe X . Instead, we define both X and \mathcal{S} as subsets of a larger ground set \mathbb{R} , and implicitly restrict \mathcal{S} to X when considering the actual set system.
- Alternatively, and perhaps more intuitively, when considering the (either weighted or unweighted) **Min-Set-Cover** problem on (X, \mathcal{S}) , we can simply imagine trying to select a min-cost subfamily of intervals in \mathcal{S} whose union is a subset of \mathbb{R} containing all of the points in X . It is clear that this is precisely equivalent to restricting \mathcal{S} to X and solving the resulting instance.
- It is indeed possible that distinct intervals in \mathcal{S} all contain the same points in X and thus become equal when restricted to X . We shall call such intervals *combinatorially equivalent*. In an instance of (possibly weighted) set cover, an optimal solution will never contain multiple combinatorially equivalent sets, and thus we can assume in all covering problems that all elements of \mathcal{S} are distinct modulo combinatorial equivalence (by deleting, if necessary, all but the cheapest set in each equivalence class).
- Similarly, we may consider two points in X to be combinatorially equivalent if they lie within the same sets in \mathcal{S} , and accordingly assume that all points in X are unique modulo combinatorial equivalence (since deleting all but one element of each equivalence class does not affect the feasibility of a potential set cover $\mathcal{C} \subseteq \mathcal{S}$).
- Even though X is permitted to be any finite subset of \mathbb{R} , we could have explicitly requested that X be a set of the form $\{1, \dots, k\}$ for some positive integer k , and requested that \mathcal{S} consist entirely of closed intervals with integer endpoints in $\{1, \dots, k\}$. It is straightforward to see that any

instance of \mathbb{R} -**Interval-SC** can be discretized in this way without affecting the element-set incidence relation. We will perform such discretization in many cases throughout this thesis.

- From the above, it follows that the number of distinct sets in \mathcal{S} is at most $\binom{M}{2} = O(M^2)$ modulo combinatorial equivalence (this exponent of 2 is related to the fact that the *VC dimension* of a \mathbb{R} -**Interval-SC** set system is at most 2; we shall explore this notion further in Section 2.4).
- It may be the case that some interval I_1 in \mathcal{S} is a sub-interval of another interval I_2 in \mathcal{S} . In the case of unweighted set cover, the smaller interval can then be ignored, since in any optimal solution, we may always assume that the larger interval can be taken. It can thus be assumed that no two intervals have the same right endpoint, and thus \mathcal{S} can be assumed to contain at most M intervals in the unweighted case. However, such simplifications are impossible in the weighted case.
- Additionally, by observing that the endpoints of \mathcal{S} cut the real number line into at most $2N + 1$ sub-intervals (of which at most $2N - 1$ are contained in at least one set in \mathcal{S}), it follows that the number of distinct points in X is at most $2N - 1 = O(N)$ modulo combinatorial equivalence.

The unweighted \mathbb{R} -**Interval-SC** problem can easily be solved in polynomial time via a straightforward left-to-right greedy algorithm in which we repeatedly take the interval that, among those covering the leftmost uncovered element, has the largest right endpoint. The weighted version can be solved in a similar manner via a dynamic programming algorithm in which $W[k]$ —the cheapest cost of covering the leftmost k points in X —is computed for all k via a simple recurrence. We leave, as a straightforward exercise for the reader, verification of the fact that these methods produce polynomial algorithms. In Chapter 4, we will generalize these methods to obtain polynomial-time algorithms for more difficult covering problems.

\mathbb{R} -**Interval-SC** set systems have a number of equivalent definitions. Here is a variant:

COVERING PROBLEM: **Path-SC**
 ELEMENTS: The set E of edges of a path graph $P = (V, E)$
 SETS: A family of paths in P

Here, by a path graph, we simply mean a path—a tree with all vertices having degree at most 2. The correspondence between \mathbb{R} -**Interval-SC** and **Path-SC** instances is obvious—points correspond to edges, and intervals covering a group of consecutive points correspond to paths covering a group of consecutive edges in P . We state **Path-SC** only for the purpose of helping to clarify the two other covering problems we mention in this section, both of which are generalizations of \mathbb{R} -**Interval-SC**. One— \mathbb{R}^2 -**Rectangle-SC**—is a geometric generalization analogous to \mathbb{R} -**Interval-SC** in higher dimensional Euclidean space. The other—**Tree-SC**—is a combinatorial generalization obtained by taking the definition of **Path-SC** and replacing ‘path graph’ with ‘tree’.

1.2.2 Rectangle Cover

We first consider a 2-dimensional analogue of \mathbb{R} -**Interval-SC**. Instead of our ground set consisting of a collection of points in \mathbb{R} , it shall consist of points in the plane. Additionally, we shall replace intervals with Cartesian products of intervals. The resulting problem, known commonly as *rectangle cover*, is the following:

COVERING PROBLEM: \mathbb{R}^2 -**Rectangle-SC**
 ELEMENTS: A finite subset of \mathbb{R}^2
 SETS: A family of axis-aligned rectangles, each of the form $[a, b] \times [c, d]$ for $a, b, c, d \in \mathbb{R}$ with $a \leq b$ and $c \leq d$

Throughout this thesis, we will encounter many *geometric set systems* described in the above manner. The elements are points in some Euclidean space, and the sets are geometric objects or regions of some kind—in this case, axis-aligned rectangles in the plane. Such problems, particularly in two and three-dimensional spaces, occur frequently in many real-world optimization problems involving objects or regions having geometric structure.

The rectangle cover problem has a rich and colourful history. \mathbb{R}^2 -**Rectangle-SC**, and its higher dimensional analogues, have many theoretical applications in computational geometry and combinatorics, as well as less obvious areas such as machine scheduling [BP10]. However, much of the notoriety of \mathbb{R}^2 -**Rectangle-SC** is due to a special case that has arisen in many industrial problems. Before discussing it, we require the following definition:

Definition 1.2.1. A *rectilinear polygon* is a connected, closed, bounded subset of \mathbb{R}^2 that can be expressed as the union of a finite number of axis-aligned rectangles.

Equivalently, we may think of rectilinear polygons as connected, closed regions whose boundaries consist entirely of horizontal and vertical line segments. We note that the above definition allows for rectilinear polygons containing holes and points whose removal disconnects the polygon. If we want to forbid such things, we may require that the polygon be *simple*—that is, having a single piecewise-linear boundary with no self-intersections. Rectilinear polygons are sometimes also called *orthogonal polygons*.

In practical applications, a common task is to assemble a given rectilinear polygon from a minimum number of (possibly overlapping) axis-aligned rectangles. Formally, we can state this problem as follows:

COVERING PROBLEM: **Rectilinear-Polygon-Cover**
 ELEMENTS: All points inside a rectilinear polygon P
 SETS: All axis-aligned rectangles lying entirely inside P

We note that, in the form stated above, **Rectilinear-Polygon-Cover** set systems (X, \mathcal{S}) have neither X nor \mathcal{S} being finite sets. We shall see later that **Rectilinear-Polygon-Cover** instances can be discretized to obtain equivalent covering problems on finite set systems.

Somewhat confusingly, the **Rectilinear-Polygon-Cover** problem sometimes also goes by the name *rectangle cover*. However, it is indeed not as general as \mathbb{R}^2 -**Rectangle-SC**, and as we shall see, its known approximability properties differ tremendously. We shall always assume that instances of **Rectilinear-Polygon-Cover** are unweighted.

The **Rectilinear-Polygon-Cover** problem shows up in a number of important industrial applications dating back several decades. Perhaps the most crucial application is to VLSI (very large scale integration) circuit design [Heg82]. In this setting, the rectilinear polygon may represent a region within a layer of a silicon microchip onto which a desired type of semiconductor material must be deposited via photolithographic techniques. A layer of polygons is printed by exposing photosensitive materials to ultraviolet light passing through a mask, which itself is created by a series of overlapping rectangular flashes of light. The cost of a mask increases with the number of flashes required, and thus obtaining a cover for the rectilinear polygon using fewer rectangles results in a reduction in the number of flashes required, decreasing production time and costs.

More recently, similar techniques have been developed for the design of DNA chip arrays used in a variety of applications such as genomic analysis [HHLPO2]. As in the VLSI circuit design example, light passing through a mask is used, but this time, the light controls the synthesis of oligonucleotides arranged in a rectangular array. As with the case of silicon chip fabrication, the goal is to minimize the number of flashes required in production of the mask, and doing so is equivalent to finding an optimal solution to a **Rectilinear-Polygon-Cover** instance.

A final application of **Rectilinear-Polygon-Cover** is to compression algorithms, particularly those used in some types of image storage. Finding a solution to a **Rectilinear-Polygon-Cover** instance enables an arbitrary configuration of pixels, such as those found in a black and white image, to be represented as a collection of rectangular blocks, possibly reducing the amount of information required to store the configuration. Such a method is used in [CIK88] to obtain competitive compression ratios for certain types of images.

At first, it may not be immediately obvious that **Rectilinear-Polygon-Cover** can be reduced to \mathbb{R}^2 -**Rectangle-SC**, as both the number of points to cover and the number of available rectangles to use appears to be infinite in a **Rectilinear-Polygon-Cover** instance. However, we can apply discretizing tricks that allow us to assume otherwise. For a rectilinear polygon P containing $2k$ edges, we can discretize P so that every vertex lies on an integer lattice point of a k by k grid, since vertices of P may have at most k distinct x -coordinates and k distinct y -coordinates. It then suffices to find a cover of the integer lattice points lying inside P by rectangles within P whose vertices have integer coordinates. There are at most k^2 such points and at most $O(k^4)$ such rectangles (in fact, only $O(k^2)$ if only maximal rectangles are considered [Fra89]), so indeed we can encode every instance of **Rectilinear-Polygon-Cover** as an instance of \mathbb{R}^2 -**Rectangle-SC** with only a polynomial blowup in problem size.

From a practical point of view, **Rectilinear-Polygon-Cover** seems like a relatively unimposing problem that admits decent approximations in the

1. INTRODUCTION

real world. Many heuristic algorithms for **Rectilinear-Polygon-Cover** have been observed to perform well on a variety of instances, both randomly generated and obtained from practical applications. In an application for DNA chip arrays, Hannenhalli, Hubbell, Lipshutz, and Pevzner devised an efficient algorithm that computed provably optimal rectilinear polygon covers for every sample instance found in their company test data. In a recent large-scale experiment by Heinrich-Litan and Lubbecke [HLL06], a relatively simple primal-dual algorithm for **Rectilinear-Polygon-Cover** was implemented and tested on a large number of randomly generated instances of various types as well as black and white test images obtained from a variety of sources. On all instances, even very large ones, the authors obtained results whose costs were provably within a few percent of optimal.

However, from a theoretical perspective, much less is known. **Rectilinear-Polygon-Cover** is quite infamous for having stumped and frustrated many combinatorialists and theoretical computer scientists over the decades. No constant approximation is known except in special cases, such as when the rectilinear polygon is simple (in which case a 2-approximation is known [Fra89]) or vertically convex (in which case a polynomial-time exact algorithm via dynamic programming is known [FK84]). For the general problem, the best known approximation is an algorithm of Anil Kumar and Ramesh that achieves an approximation factor of $O(\sqrt{\log n})$ on instances having n edges [AKR03]. From the hardness side, **Rectilinear-Polygon-Cover** is known to be **MAX-SNP-hard**, implying that no polynomial time approximation scheme (PTAS) exists unless $\mathbf{P} = \mathbf{NP}$ [BD97]. Additionally, the problem remains **NP-hard** even when the polygon is simple [CR94]. However, a constant approximation has not been ruled out, and finding one (or disproving the existence of one) remains a key open problem in the area.

The approximability of **Rectilinear-Polygon-Cover** is intimately related to a longstanding open problem in combinatorics involving rectilinear polygons. For a rectilinear polygon P , if I is a set of points in P having the property that no two points in I are contained in any rectangle lying entirely within P , then it is clear that at least $|I|$ rectangles are needed to cover P . Such a set I is called an *independent set*, or sometimes a *stable set* or *antirectangle*. If θ is the size of a minimum rectangle covering of P and α is the size of a maximum independent set in P , then we must always have $\theta \geq \alpha$ by our reasoning above. However, it is not obvious whether θ can ever exceed α by much; in fact, quite a remarkable story surrounds the quest to understand the so-called *packing-covering duality gap* θ/α (see, e.g. [CKSS81]). Chvátal once conjectured that θ and α were equal for all rectilinear polygons, but Szemerédi produced a counterexample sometime in the seventies. Chung, in 1979, gave the first counterexample containing no holes, proving that θ/α can be at least as large as $\frac{8}{7}$, even for simple polygons. As of the time of writing, it appears that no better counterexample has ever been published.⁷ Erdős asked if θ/α can be bounded above by *any* constant.

⁷Chaiken et al. claimed, in a 1981 paper, to have found an example having a larger gap of $\frac{21}{17} - \epsilon$ [CKSS81]. However, they did not publish their example and other authors report

Despite many partial results for special cases (e.g. [CKSS81, Lub85, Lub90]), the question appears to remain unresolved.

Finding a constructive proof of a constant upper bound on θ/α could lead to a primal-dual-based constant approximation for **Rectilinear-Polygon-Cover**, but progress seems to have stalled in recent decades. An interesting and more modern approach to such problems is the use of topological methods—in particular, application of existential results such as the Brouwer fixed-point theorem and the Borsuk-Ulam theorem (a book by Matoušek [Mat03] contains a comprehensive overview of the relevant techniques). Topological methods involving these theorems have yielded existence proofs of optimal packing-covering duality gaps for similar geometric set systems [Tar95, Kai97]. Classical proofs of these results are not known. A noteworthy drawback of topological techniques is that even if they succeed in establishing upper bounds on packing-covering duality gaps, they may not lead to polynomial time approximation algorithms, as they are inherently *non-constructive*.

Despite the wealth of classical combinatorial results concerning **Rectilinear-Polygon-Cover**, somewhat less is known about the general \mathbb{R}^2 -**Rectangle-SC** problem, which could potentially be more difficult from a theoretical inapproximability point of view. No algorithm is known to achieve an $o(\log M)$ -approximation on an instance having M points; it thus remains possible that \mathbb{R}^2 -**Rectangle-SC** admits no better approximation than the general **Min-Set-Cover** problem. Despite this, the strongest inapproximability result known for \mathbb{R}^2 -**Rectangle-SC** is **APX**-hardness [vL09]; in particular, no super-constant hardness results are known.

A very recent and exciting development is a construction due to Pach and Tardos yielding an instance of \mathbb{R}^2 -**Rectangle-SC** in which the integrality gap of the natural linear programming relaxation (SCLP) is $O(\log M)$ [PT11]. While stopping short of obtaining any formal proof of inapproximability, their result has ruled out many potential solution techniques, including all those that would achieve an approximation ratio relative to the optimal value of (SCLP). However, their result cannot be generalized to provide any improved linear programming duality gap examples for **Rectilinear-Polygon-Cover**; therefore, nothing has changed regarding Erdős’s question on the packing-covering duality gap of **Rectilinear-Polygon-Cover** instances. Further ramifications of the Pach-Tardos construction are discussed in Section 3.3.

Unfortunately, we obtain no new results for the general \mathbb{R}^2 -**Rectangle-SC** or **Rectilinear-Polygon-Cover** problems in this thesis. However, we do study several special cases and related problems, obtaining many improved algorithmic and hardness results. In particular:

- In Chapter 2, we obtain structural results on concerning a wide class of covering problems similar to \mathbb{R}^2 -**Rectangle-SC**.
- In Chapter 4, we give polynomial-time exact algorithms for several weighted

being unable to locate or reconstruct it [HLL06].

1. INTRODUCTION

covering problems, including \mathbb{R}^2 -**Rectangle-SC** instances in which all input rectangles are about the x -axis.

- In Chapter 5, we prove **APX**-hardness for various unweighted geometric covering problems, including \mathbb{R}^2 -**Rectangle-SC** when all rectangles in the input are ϵ -perturbed copies of a single unit square, or when all rectangles are either horizontal or vertical slabs.

1.2.3 Tree Cover

Another interesting problem can be obtained by generalizing \mathbb{R} -**Interval-SC** via a combinatorial route rather than a geometric one. By taking the equivalent **Path-SC** problem and generalizing paths to trees, we obtain a problem known as *Tree Cover*:

COVERING PROBLEM: **Tree-SC**
 ELEMENTS: The set E of edges of a tree $T = (V, E)$
 SETS: A family of paths in T

The **Tree-SC** problem arises in a variety of contexts in combinatorial optimization. One such instance is the so-called *Tree Augmentation* problem: given is a tree T and a set S of non-tree edges whose endpoints are vertices of T , and our goal is to select of a minimum cardinality (or minimum cost) subset of S whose addition to T augments the edge connectivity of T by one. In other words, we must find a minimum $C \subseteq S$ such that $T \cup C$ is *two-edge-connected*. To see the equivalence between **Tree-SC** and Tree Augmentation, observe that adding any $e = (u, v) \in S$ to T will create a cycle (known as a *fundamental cycle*) containing the path from u to v in T , and that $T \cup C$ will be *two-edge-connected* if and only if each edge of T lies within at least one such cycle. It follows that finding a solution to a Tree Augmentation instance is equivalent to solving a **Tree-SC** instance in which each path in P corresponds to the tree edges lying in a single fundamental cycle. One may also observe that the problem of augmenting the connectivity of a general connected graph to 2 is equivalent to the Tree Augmentation problem after contracting all cycles; this type of problem is sometimes simply called *Bi-connectivity Augmentation*.

An entirely different but equivalent framework for the **Tree-SC** problem is the so-called *Laminar Cover* problem.

Definition 1.2.2. A family \mathcal{L} of sets is said to be *laminar* if for all $X, Y \in \mathcal{L}$ we have either $X \subseteq Y$, $Y \subseteq X$, or $X \cap Y = \emptyset$.

Given a graph $G = (V, E)$ and a laminar family \mathcal{L} on V , an edge $e = (u, v) \in E$ is said to *cross* a set $S \in \mathcal{L}$ whenever $u \in S$ and $v \notin S$ or $u \notin S$ and $v \in S$. Given such a G and \mathcal{L} , the *Laminar Cover* problem asks us to find a minimum cardinality subset $C \subseteq E$ such that every set in \mathcal{L} is crossed at least once by an edge in C . Noting that the sets in a laminar family form a tree-like structure under the inclusion relation, it is not hard to see that the *Laminar Cover* problem is again equivalent to **Tree-SC**.

Both weighted and unweighted **Tree-SC** have a number of useful applications to the solution of various network design problems (see, e.g. [Jai01], or the survey [KN07]). The goal in these types of problems is typically to build or purchase a network meeting certain connectivity or reliability requirements.

Tree-SC is **NP-hard**, even in the unweighted case, and even when T has diameter four [FJ82]. However, a noteworthy special case admits a polynomial-time exact algorithm:

COVERING PROBLEM: **Vertical-Tree-SC**
 ELEMENTS: The set E of edges of a rooted tree $T = (V, E)$
 SETS: A family of vertical paths in T

Here, by a vertical path in a rooted tree T , we mean a path from any vertex v in T to an ancestor or descendent of v .

Weighted **Vertical-Tree-SC** can be solved exactly in polynomial time via a dynamic programming algorithm that recursively computes the minimum cost of covering all of the edges in each subtree of T . However, it turns out that the set system matrices induced by **Vertical-Tree-SC** are *totally unimodular* [Sch03], implying that the solution obtained via the natural linear programming relaxation (SCLP) is an exact integral solution, in turn providing an entirely different polynomial algorithm for **Vertical-Tree-SC**.

The polynomial solvability of **Vertical-Tree-SC** can be used to obtain an easy 2-approximation for general **Tree-SC** [FJ82]. The key observation we need is the following elementary result:

Proposition 1.2.3. *If P is an arbitrary path in a rooted tree T , then P can be split into the disjoint union of two (possibly empty) paths L and R that are each vertical paths in T (here, L and R represent the left and right vertical pieces of P).*

The method for obtaining a 2-approximation for general **Tree-SC** is then:

-
- 1: Input a (possibly weighted) **Tree-SC** instance on a tree T .
 - 2: Select an arbitrary vertex v of T and consider T to be rooted at v .
 - 3: Apply Proposition 1.2.3 to each path P_j in \mathcal{S} to replace it with paths L_j and R_j in T that are each vertical with respect to v .
 - 4: Assign L_j and R_j the same weight as P_j , and obtain an exact optimal solution to the auxiliary **Vertical-Tree-SC** instance containing only the paths in $\{L_j \cup R_j : j \in \{1, \dots, N\}\}$.
 - 5: Output a solution to the **Tree-SC** instance in which a path P_j is taken if and only if at least one of L_j or R_j was taken in the optimal solution to the auxiliary **Vertical-Tree-SC** instance.
-

It is clear that the cost of the **Vertical-Tree-SC** solution obtained in Algorithm 1.2.3 is at most twice the cost of an optimal **Tree-SC** solution, and from this, we obtain the following result:

Theorem 1.2.4. *Weighted **Tree-SC** admits a 2-approximation.*

1. INTRODUCTION

For nearly twenty years, the best known approximation factor for **Tree-SC** stood at 2, and improving upon this was regarded as a major open problem in the area [Khu97]. It was not until 1999 that a better ratio was found; then, Nagamochi gave a $(1.875 + \epsilon)$ -approximation for *unweighted Tree-SC* [Nag03]. This was subsequently improved to a 1.5-approximation by Even et al. in 2001 [EFKN01], which remains the best approximation factor for the general problem. However, improved results are known for special cases, such as when T has low diameter, or when all the paths in \mathcal{S} are leaf-to-leaf paths [Mad09]. For *weighted Tree-SC*, it is still unknown whether there exists any algorithm that achieves a better approximation than the 2-factor provided by Algorithm 1.2.3.

Most algorithms for **Tree-SC** make use of very specific combinatorial properties of **Tree-SC** set systems, and the more general methods we develop in this thesis are not able to yield improved results. However, we obtain a variety of interesting results relating to many variations and generalizations of **Tree-SC** and **Vertical-Tree-SC**. Some examples:

- In Chapter 5, we prove that the *priority* version of **Vertical-Tree-SC** is **APX**-hard.
- In Chapters 6 and 7, we give an algorithm that yields a constant approximation for weighted capacitated and priority versions of **Tree-SC**.
- In Section 2.3, we give a direct encoding reduction showing that **Vertical-Tree-SC** is in fact a special case of **\mathbb{R}^2 -Rectangle-SC**.

1.3 Organization of Thesis

This thesis comprises eight chapters. Chapter 2 develops all of the necessary preliminaries. Included are descriptions of several set systems alongside structural results that relate and characterize them. We formally define the reduction-based partial order defined by *subproblems*; this allows us to formally show that some covering problems generalize others. We prove many relations under this ordering, some of which we believe were not previously known. We also discuss measures of *set system complexity*, the most important of which is *Vapnik-Chervonenkis dimension* (VC dimension). Set systems of low VC dimension often admit strong approximations via algorithms known as *ϵ -net-finders*; we explain all of the necessary background information required to understand this.

The remainder of the thesis contains a collection of algorithmic and hardness results for covering problems. Most of the problems analyzed herein can be placed into one of two categories—those like **\mathbb{R}^2 -Rectangle-SC** that arise from geometric applications, and those like **Tree-SC** that correspond to combinatorial optimization problems in graphs.⁸ There is a great deal of prior work

⁸Of course, there is some overlap in cases such as **\mathbb{R} -Interval-SC** for which formulations of both types exist.

1.3. ORGANIZATION OF THESIS

concerning these types of problems, and we summarize and highlight the important contributions in Chapter 3. Results are organized according to the types of techniques employed.

Chapters 4 through 6 contain original results, most of which are from articles that appear in (or have been submitted to) refereed publications. Chapter 4 contains dynamic programming algorithms for several weighted geometric covering problems, proving that they are exactly solvable in polynomial time. Results using this method were first published in [CGK10a], and subsequently improved in [CG11]. Chapter 5 contains a collection of **APX**-hardness proofs for several covering problems. Most of these results were originally given in [CG11]. All of them make use of the **APX**-hardness of a very specific covering problem we call **Special-3SC**. Chapter 6 describes an algorithm from [CGKS12] for weighted covering problems based on a *quasi-uniform sampling* technique. It produces an output whose approximation guarantee varies with an instance-specific parameter known as *shallow cell complexity*. Several new results for weighted covering problems are achieved using this method. Finally, in Chapter 7, we apply our results to various *capacitated* and *priority* covering problems to achieve improved results. A major contribution is an algorithm based on linear programming strengthening and rounding, originally published in [CGK10a] (see also [CGK10b]).

In Chapter 8, we summarize all the results discussed and present a listing of all the problems studied in this thesis, classified by their approximability (where known). We also discuss some directions for future research.

Chapter 2

Foundations

In this chapter, we cover the preliminary definitions and basic results for the types of set systems we shall study. The key ideas focus around characterizing different types of structure present in set systems, and using this information to obtain algorithms for set cover problems, or to relate set cover problems to one another. Many of the results in this section are trivial, or are folklore-type results that are well known, even if rarely stated explicitly.

2.1 Set System Matrices

We recall that a set system can be defined by a binary M by N *set system matrix* A , whose rows are indexed by elements, whose columns are indexed by sets, and whose equivalence with other such matrices is modulo row and column reorderings. Since a covering problem can be regarded as a class of set systems, it accordingly can be regarded as a class of binary matrices.

In this thesis, we shall often take the viewpoint of stating properties of covering problems as properties of their respective set system matrices. For a more natural presentation, we will simply equivocate classes of set systems with classes of set system matrices instead of fussing over the details of the two representations (matrices A and set systems (X, \mathcal{S})). We choose to employ the matrix-based terminology instead of the set-system-based terminology in the vast majority of cases. In particular, our prose shall discuss the rows and columns of a set system matrix A rather than the elements X and sets \mathcal{S} in a set system (X, \mathcal{S}) . Although this may seem unconventional (particularly to computational geometers), we believe that it aids in distilling useful structure down to fundamental combinatorial properties and results in a shorter, cleaner presentation of many results (especially in Chapter 6). In a sense, everything we need to know about the structure of classes of set systems shall boil down to which types of submatrices they admit or forbid.

We spend the remainder of this section developing the vocabulary surrounding set systems and their matrix representations. Proofs of trivial results are

omitted if they follow immediately from definitions.

2.1.1 Duality

Recall that, given a set system (X, \mathcal{S}) , its *dual set system* $(X, \mathcal{S})^*$ is the set system obtained by taking (X, \mathcal{S}) and interchanging the role of elements and sets. Duality has a natural formulation in terms of set system matrices:

Proposition 2.1.1. *If $A = A_{(X, \mathcal{S})}$ is the set system matrix for (X, \mathcal{S}) , then its transpose A^T is the set system matrix¹ for $(X, \mathcal{S})^*$.*

This enables us to extend our definition of duality to classes of set systems, and in turn, to set cover problems:

Definition 2.1.2. For a class \mathbf{C} of set systems, we will write \mathbf{C}^* for the class $\{A^T : A \in \mathbf{C}\}$.

We summarize the correspondence between primal and dual set systems in Table 2.1.1.

	Primal	Dual
Set system	(X, \mathcal{S})	$(X, \mathcal{S})^*$
Set system matrix	A	A^T
Class of set systems	\mathbf{C}	\mathbf{C}^*

Table 2.1: Relation between primal and dual set systems

2.1.2 Heredity

A noteworthy property of many covering problems is that deleting elements or sets produces a covering problem of the same type:

Definition 2.1.3. A class \mathbf{C} of set systems is *hereditary* if for all set system matrices $A \in \mathbf{C}$, we have $B \in \mathbf{C}$ whenever B is a submatrix of A (equivalently, membership in \mathbf{C} is unchanged when rows or columns of A are deleted). A covering problem is *hereditary* whenever its corresponding class of set systems is.

The vast majority of covering problems we examine in this thesis are hereditary. For example:

¹Recalling that we always consider set system matrices *modulo row and column reorderings*, and noting that the matrix transpose operation commutes with reordering rows and columns (e.g., performing a row swap and then transposing is equivalent to transposing and then performing the equivalent column swap), we observe that the set system matrix A^T is in fact well-defined.

- \mathbb{R}^2 -**Rectangle-SC** is hereditary, because removing points or rectangles from a \mathbb{R}^2 -**Rectangle-SC** instance produces another \mathbb{R}^2 -**Rectangle-SC** instance.
- Analogously, the same thing holds for all other geometric set cover problems where the elements are points and the sets are geometric objects of some kind.
- **Tree-SC** is also hereditary, but it is perhaps less obvious deleting a row from a **Tree-SC** matrix shall always leave a **Tree-SC** matrix; to see this, note that deleting a row of a **Tree-SC** matrix A corresponds to contracting an edge in the corresponding tree T .
- An example of a covering problem that is *not* hereditary is **Rectilinear-Polygon-Cover**.

It is clear that the hereditary property is preserved when computing the dual of a class of set systems:

Proposition 2.1.4. *If a class \mathbf{C} of set systems is hereditary, then so is \mathbf{C}^* .*

2.1.3 Subproblems

Despite being trivial, the following definition is extremely important:

Definition 2.1.5. Let \mathbf{C}_1 and \mathbf{C}_2 be two classes of set systems. We shall write $\mathbf{C}_1 \subseteq \mathbf{C}_2$ and say that \mathbf{C}_1 is a *subproblem* of \mathbf{C}_2 or \mathbf{C}_2 *encodes* \mathbf{C}_1 if, for all matrices $A \in \mathbf{C}_1$, we also have $A \in \mathbf{C}_2$.

We shall employ the same vocabulary and notation for covering problems. For example, we shall say that \mathbb{R} -**Interval-SC** is a subproblem of \mathbb{R}^2 -**Rectangle-SC** and write \mathbb{R} -**Interval-SC** \subseteq \mathbb{R}^2 -**Rectangle-SC**.

The subproblem relation ‘ \subseteq ’ shall prove crucial in allowing us to relate different covering problems, for it tells us precisely when one problem is a special case of another via a direct instance-to-instance mapping. We observe the following:

Remark 2.1.6. The subproblem relation ‘ \subseteq ’, when applied to covering problems, is transitive, antisymmetric, and reflexive. Therefore, the set of all covering problems forms a *partially ordered set (poset)* with respect to the ‘ \subseteq ’ relation.

The following are a few easy examples discussed in the introduction:

- \mathbb{R} -**Interval-SC** \subseteq \mathbb{R}^2 -**Rectangle-SC**
- **Rectilinear-Polygon-Cover** \subseteq \mathbb{R}^2 -**Rectangle-SC**
- \mathbb{R} -**Interval-SC** \subseteq **Vertical-Tree-SC**
- **Vertical-Tree-SC** \subseteq **Tree-SC**

2. FOUNDATIONS

We give many other (less obvious) encodings in Section 2.3.

It turns out that the subproblem relation commutes with the duality operator:

Proposition 2.1.7. *Let \mathbf{C}_1 and \mathbf{C}_2 be two classes of set systems. Then $\mathbf{C}_1 \subseteq \mathbf{C}_2$ if and only if $\mathbf{C}_1^* \subseteq \mathbf{C}_2^*$.*

Proof. The result follows immediately from the fact that $A \in \mathbf{C}$ if and only if $A^T \in \mathbf{C}^*$. \square

2.1.4 The Integrality Gap

We recall the integer programming formulation for a (possibly weighted) covering problem, and its natural linear programming relaxation:

$$\min\{\mathbf{w}^T \mathbf{x} : A\mathbf{x} \geq \mathbf{1}, \mathbf{x} \in \{0, 1\}\} \quad (\text{SCIP})$$

$$\min\{\mathbf{w}^T \mathbf{x} : A\mathbf{x} \geq \mathbf{1}, \mathbf{x} \geq \mathbf{0}\} \quad (\text{SCLP})$$

Denote by $OPT(A, \mathbf{w})$ the optimal value of the integer program (SCIP) on a set system A with weights \mathbf{w} . Let $OPT_f(A, \mathbf{w})$ be the optimal value of the linear programming relaxation (SCLP) on the same input; in other words, let $OPT_f(A, \mathbf{w})$ be the cost of the optimal *fractional* set cover of a set system A with weights \mathbf{w} . The quantity $OPT_f(A, \mathbf{w})$ provides a simple lower bound on the cost of an optimal solution, and this can be used to obtain approximation guarantees for algorithms:

Definition 2.1.8. An algorithm for a set cover problem is an *LP-relative α -approximation* if it always produces a solution of value at most $\alpha OPT_f(A, \mathbf{w})$ over all choices of A (and in the weighted case, over all $\mathbf{w} \in \mathbb{R}_+^N$).

The ratio of $OPT(A, \mathbf{w})$ to $OPT_f(A, \mathbf{w})$ provides a measure of how well the linear programming relaxation approximates the optimal solution to a single instance. When we examine this quantity over a class of instances and look at the worst case, we gain information on how tight the relaxation may be for a given set cover problem:

Definition 2.1.9. The *weighted integrality gap* of a class of set systems \mathbf{C} is given by

$$\sup_{A \in \mathbf{C}, \mathbf{w} \in \mathbb{R}_+^N} \frac{OPT(A, \mathbf{w})}{OPT_f(A, \mathbf{w})}.$$

The *unweighted integrality gap* of \mathbf{C} is given by

$$\sup_{A \in \mathbf{C}} \frac{OPT(A, \mathbf{1})}{OPT_f(A, \mathbf{1})}.$$

Additionally, the integrality gap of a (weighted or unweighted) covering problem is the (weighted or unweighted, respectively) integrality gap of its corresponding class of set systems.

Note that we might also express integrality gaps (sometimes also called *integrality ratios*) as functions of N ; as an example, the general unweighted **Min-Set-Cover** problem is known to have an integrality gap of $O(\log N)$, since

$$OPT(A, \mathbf{w}) \leq O(\log N) \cdot OPT_f(A, \mathbf{w})$$

for all instances (where N is the number of columns of A) [Vaz01]. As it turns out, this is asymptotically tight (the integrality gap of **Min-Set-Cover** is $\Theta(\log N)$); see [Vaz01] for an example).

We will sometimes attempt to compute or bound the integrality gaps of various covering problems. The integrality gap may not necessarily tell us explicitly what the best approximability or inapproximability for a problem is, but it can provide information about what is achievable with certain methods. For example, the following is clear:

Proposition 2.1.10. *Suppose a class of set systems \mathbf{C} has weighted (or respectively, unweighted) integrality gap α . Then the weighted (respectively, unweighted) set cover problem on \mathbf{C} has no LP-relative β -approximation for any $\beta < \alpha$.*

This will prove crucial when we discuss ϵ -net finders (see Section 2.4), because their existence implies an LP-relative approximation, and thus examples having large integrality gap can rule out the existence of ϵ -net-based methods for certain problems (see Section 3.3 for a more thorough discussion and example).

2.1.5 Total Unimodularity

One case where the integrality gap of a set system can easily be determined is when the set system matrix A is *totally unimodular*. We recall the following classic definition:

Definition 2.1.11. An integer matrix A is *totally unimodular* (TUM) whenever every square submatrix of A has a determinant in $\{0, 1, -1\}$. Equivalently, A is TUM if and only if every square submatrix of A is either singular or has an integer inverse.

We shall also say that a set system is TUM whenever its corresponding set system matrix is.

Total unimodularity has important consequences for the covering integer program (SCIP). For example, the following can easily be deduced from classical results (see, e.g. Chapter 83 of [Sch03] or Chapter 19 of [Sch86] for justification):

Theorem 2.1.12. *Suppose A is TUM and let \mathbf{b} be a vector of integers. Then the polyhedron $\{\mathbf{x} \geq \mathbf{0} \mid A\mathbf{x} \geq \mathbf{b}\}$ is integral. Accordingly, the linear program (SCLP) has an integral optimal solution for any weights \mathbf{w} .*

This theorem implies that the value of an optimal solution to (SCIP) is the same as the value of an optimal solution to (SCLP) whenever A is TUM. In other words, we have the following:

2. FOUNDATIONS

Corollary 2.1.13. *Let \mathbf{C} be a class of totally unimodular set systems. Then \mathbf{C} has an integrality gap of 1.*

Moreover, we can solve (SCIP) (and consequently, the minimum set cover problem on \mathbf{C}) exactly by simply finding an optimal basic feasible solution to (SCLP). Since the optimal solution of such a linear program can be computed in polynomial time (e.g. via the ellipsoid method [Sch86]), we obtain the following:

Corollary 2.1.14. *Let \mathbf{C} be a class of totally unimodular set systems. Then the minimum cost set cover problem on \mathbf{C} can be solved exactly in polynomial time.*

To make use of this, we require some examples of TUM set systems. We begin with the following:

Proposition 2.1.15. *Let A be TUM. Then A^T is TUM. Accordingly, if \mathbf{C} is a totally unimodular set system class, then so is \mathbf{C}^* .*

Proof. This follows immediately from Definition 2.1.11 and the fact that the determinant of A is equal to the determinant of A^T . □

An important class of TUM matrices are the *network matrices*:

Definition 2.1.16. An M by N matrix A is a *network matrix* if and only if there exists a directed tree T on edges $\{e_1, \dots, e_M\}$ and a family of directed paths $\{P_1, \dots, P_N\}$ in T such that:

- $A_{i,j} = 1$ whenever e_i lies along P_j in a forward direction.
- $A_{i,j} = -1$ whenever e_i lies along P_j in a backward direction.
- $A_{i,j} = 0$ whenever e_i does not lie in P_j .

We can define **Network-SC** as the covering problem consisting of all set cover matrices that are network matrices (in other words, all network matrices that have no entries equal to -1). This is precisely equivalent to the following:

COVERING PROBLEM: **Network-SC**
 ELEMENTS: The set of edges of a directed tree T
 SETS: A family of directed paths in T , each of which respects the directions of edges in T

It follows immediately that the **Network-SC** problem is a generalization of **Vertical-Tree-SC** and thus

$$\mathbf{Vertical-Tree-SC} \subseteq \mathbf{Network-SC}.$$

Additionally, immediately from Definition 2.1.16 we have:

Proposition 2.1.17. *The problem **Network-SC** is hereditary. That is, every submatrix of a binary network matrix is also a binary network matrix.*

The following is a classical result (see, e.g., Section 13.3 of [Sch03]):

Theorem 2.1.18. *Let A be a network matrix. Then A is totally unimodular.*

By applying Corollaries 2.1.13 and 2.1.14, we obtain:

Corollary 2.1.19. *The covering problems \mathbb{R} -**Interval-SC**, **Vertical-Tree-SC**, and **Network-SC** all have integrality gap 1 and are thus solvable in polynomial time.*

Practically speaking, when TUM matrices manifest in a combinatorial optimization context, they often turn out to also be network matrices (or their transposes). In fact, it is uncommon to stumble upon matrices that are TUM but are *not* network matrices. This occurs for a profound reason; in a classic and celebrated result spanning more than 50 pages, Seymour showed that all TUM matrices admit a certain type of construction from a class of atomic units comprising network matrices, their transposes, and a single sporadic 5 by 5 matrix [Sey80]. Consequently, the structure of TUM matrices is very closely related to that of network matrices.²

2.2 Complexity-Theoretic Preliminaries

Here, we give brief definitions of the complexity classes and reductions that we will use in this thesis, and explanations of how these notions relate to covering problems. We state only the results that we shall explicitly use in the forthcoming chapters. For more details, including more rigorous definitions and proofs of everything described here, one may refer to any of a number of standard texts on the subject such as [AB09].

The class \mathbf{P} is commonly used for the set of decision problems that can be solved by a deterministic Turing machine in polynomial time. In this thesis, we take the standard approach of considering optimization problems to be in \mathbf{P} whenever their corresponding decision problems are. For unweighted covering problems or covering problems with explicitly represented integer weights, this holds if and only if there exists a polynomial algorithm that returns the optimal objective value, since the optimal value can be found in a polynomial number of steps via binary searching whenever the corresponding decision problem can be solved in polynomial time.³ Moreover, for all polynomial-time solvable covering problems discussed in this thesis, we shall provide algorithms that not only return the value of an optimal solution, but indeed can be easily modified to output optimal coverings.

²The decomposition of TUM matrices is intimately linked to matroid theory because of correspondences between TUM matrices and *regular matroids*, with a similar correspondence holding for network matrices and *graphic matroids*. We will not discuss matroid theory in this thesis, but the interested reader is eagerly referred to Truemper's book [Tru92] for a thorough presentation of Seymour's decomposition theorem and all of its implications.

³In fact, this holds for all optimization problems in which all feasible objective values are sums or products of explicitly represented rational numbers [Sch03].

2. FOUNDATIONS

The decision version of the general **Min-Set-Cover** problem can be easily seen to be in **NP**, since the covering itself provides a polynomial-time checkable certificate that a covering of a certain size exists. Consequently, in this thesis, we shall proceed under the assumption that $\mathbf{P} \neq \mathbf{NP}$, since otherwise, all covering problems would be exactly solvable in polynomial time, and most of our approximability bounds would be meaningless. From the hardness side of things, the decision version of general **Min-Set-Cover** is also **NP-hard** [Kar72] and is therefore an **NP-complete** problem. We shall say that an optimization problem is **NP-hard** whenever its corresponding decision version is. Many more restricted covering problems such as **\mathbb{R}^2 -Rectangle-SC** and **Tree-SC** are also **NP-hard** (see Section 3.1 for examples).

For the majority of this thesis, we shall not be overly concerned with the run time of our algorithms and shall usually just prove that they terminate in time polynomial in the input size of the problem. For covering problems, the input may either be an explicit matrix containing MN binary entries, or a more compact representation such as a list of locations of points and objects defining a geometric set system. When polynomial time is all that we care about, the distinction between these input representations is unimportant since a function is polynomial in MN if and only if it is polynomial in a smaller function in $\Omega(M + N)$. Accordingly, we shall ignore the details of input storage for the covering problems discussed herein. Only in Chapter 4 do we explicitly provide worst-case run time bounds, and in this case, we simply express them directly in terms of M and N assuming an appropriate set of primitive operations.

Many of our results deal with approximation algorithms. We shall use the following generalized approximation ratios:

Definition 2.2.1. Given a covering problem \mathcal{P} , an algorithm is an $f(M, N)$ -*approximation* for \mathcal{P} if, for all M and N , it always produces solutions with costs at most $f(M, N)$ times optimal on instances in \mathcal{P} having set systems with M elements and N sets.

This definition enables us to define approximation guarantees that vary only with M or N rather than the entire problem input size. Of course, wherever possible, we shall aim for approximation algorithms for which $f(M, N)$ is simply a constant; so-called *constant approximations* or *constant factor approximations* are obtainable for a variety of covering problems. Problems admitting these make up the complexity class **APX**:

Definition 2.2.2. A covering problem \mathcal{P} belongs to the class **APX** if, for some $\alpha \geq 0$, \mathcal{P} admits a $(1 + \alpha)$ -approximation that runs in polynomial time.

One example of a problem in **APX** that we have already encountered is (weighted) **Tree-SC**.

For problems in **APX**, we shall attempt, wherever possible, to give the best possible approximation we can. Some problems are indeed **NP-hard** but admit polynomial-time $(1 + \epsilon)$ -approximations for all non-zero values of ϵ . Such problems are said to admit a *polynomial-time approximation scheme* or **PTAS**—a series of algorithms that each approximate the optimum solution with successively

improving accuracy, generally at the expense of an increase in the exponent in the algorithms' polynomial run time. Problems admitting such a scheme belong to a complexity class of their own:

Definition 2.2.3. A covering problem \mathcal{P} belongs to the class **PTAS** if, for each $\epsilon > 0$, \mathcal{P} admits a $(1 + \epsilon)$ -approximation that runs in polynomial time (where, of course, a different algorithm may be used for each value of ϵ).

The definitions of **PTAS** and **APX** essentially only differ by the replacement of an existential quantifier with a universal one.

We have the following chain of inclusions:

$$\mathbf{P} \subset \mathbf{PTAS} \subset \mathbf{APX} \subset \mathbf{NP}$$

Noteworthy is the fact that under the assumption that $\mathbf{P} \neq \mathbf{NP}$, all of these containments can be proven to be strict. Indeed, in this thesis, we will review dozens of covering problems, placing some into each of the following four types:

- Problems in **P**.
- Problems that admit a PTAS but are **NP**-hard.
- Problems that admit a constant approximation but have no PTAS.
- Problems that admit no constant approximation.

One may wonder how it might be possible to show that a covering problem admits no PTAS. Under the assumption that $\mathbf{P} \neq \mathbf{NP}$, it is widely known that there exist many problems in **APX** that are provably not in **PTAS**. An important example of a covering problem with this property is unweighted minimum vertex cover in graphs:

COVERING PROBLEM: **Min-Vertex-Cover**
 ELEMENTS: The edges of a graph $G = (V, E)$
 SETS: $\{\delta(v) : v \in V\}$, where $\delta(v)$ is the set of edges incident to v

Min-Vertex-Cover was proven to admit no approximation with factor smaller than 1.1666 [Hås99], and subsequently 1.3606 [DS05], using techniques based on the PCP theorem.⁴ Knowing this, it then suffices to construct some kind of *approximability-preserving* reduction from **Min-Vertex-Cover** to our target problem \mathcal{P} in order to prove that \mathcal{P} has no PTAS. There are two main types of reductions that accomplish this: *PTAS reductions* and *L-reductions* ('linear' reductions). Of the two, L-reductions are much simpler and are exclusively what we shall use in this thesis:

⁴The inapproximability bound of **Min-Vertex-Cover** can be improved to 2 if the *unique games conjecture* (UGC) is assumed, matching the trivial 2-approximation obtained by taking both vertices incident to every edge chosen in a maximum matching [KR08]. However, the UGC remains unproven and there appears to be no widespread consensus among experts regarding its truth. We do not require the UGC for any of the results we prove herein.

2. FOUNDATIONS

Definition 2.2.4. A pair of functions (f, g) is an *L-reduction* from a minimization problem \mathcal{A} to a minimization problem \mathcal{B} if there are positive constants α and β such that for each instance x of \mathcal{A} , the following hold:

- (L1) The function f maps instances of \mathcal{A} to instances of \mathcal{B} such that $\text{OPT}(f(x)) \leq \alpha \cdot \text{OPT}(x)$.
- (L2) The function g maps feasible solutions of $f(x)$ to feasible solutions of x such that $c_x(g(y)) - \text{OPT}(x) \leq \beta \cdot (c_{f(x)}(y) - \text{OPT}(f(x)))$, where c_x and $c_{f(x)}$ are the cost functions of the instances x and $f(x)$ respectively.

PTAS reductions are similar but slightly more involved; their definition generalizes the linear parameters α and β to allow for non-constant error blowup.

For L-reductions as well as the more general PTAS reductions, the following *approximability-preserving* properties can be shown:

Proposition 2.2.5. Write $\mathcal{A} \leq \mathcal{B}$ if there exists an L-reduction or a PTAS-reduction from \mathcal{A} to \mathcal{B} . Then:

- If $\mathcal{A} \leq \mathcal{B}$ and $\mathcal{B} \in \mathbf{APX}$, then $\mathcal{A} \in \mathbf{APX}$.
- If $\mathcal{A} \leq \mathcal{B}$ and $\mathcal{B} \in \mathbf{PTAS}$, then $\mathcal{A} \in \mathbf{PTAS}$.

These results are also extremely useful when applied in the contrapositive. Specifically, we may rule out membership in **PTAS** by constructing, e.g., an L-reduction *from* the problem **Min-Vertex-Cover**.

More general theory has also been developed. Crescenzi and Panconesi showed that the class **APX** admits a natural class of ‘complete’ problems under PTAS-reductions, analogous to the **NP**-complete problems that exist for **NP** under polynomial-time reductions [CP91]. Independently, Papadimitriou and Yannakakis showed something similar for a class known as **MAX-SNP** under L-reductions [PY91]; **MAX-SNP**-complete problems include MAX-3SAT, which admits no PTAS by the PCP theorem. As it turns out, **APX** contains **MAX-SNP** and is a natural closure of it under PTAS-reductions [CT00]. Since there are problems in **APX** that do not admit a PTAS, Proposition 2.2.5 implies that APX-hard problems also fail to admit a PTAS. **Min-Vertex-Cover** is such a problem.

In summary, for our purposes, it shall suffice to know the following:

- If a covering problem \mathcal{P} is **APX**-hard or **MAX-SNP**-hard, then it does not have a PTAS.
- We can prove that a problem is **APX**-hard by constructing an L-reduction to it from another **APX**-hard problem.
- **Min-Vertex-Cover** is **APX**-hard.

As a final note, we shall point out that many of the classical approximation algorithms for covering problems, as well as the algorithm we give in Chapter 6, employ *randomization*. However, in this thesis, we shall not concern

ourselves with the complexity-theoretic consequences of randomization, and we will not make an effort to distinguish approximability bounds that require randomization from those that do not. As we discuss in Chapter 6, it is likely that almost all known sampling-based randomized algorithms for covering problems can be derandomized in polynomial time using standard methods without any complexity-theoretic assumptions.

2.3 Set Systems and their Structural Properties

Here, we shall explain various ways of constructing set systems having exploitable structural properties. Many of these examples are obtained by taking existing set systems and applying various operations to them to create new set systems. We will exploit the properties of these set systems to obtain improved approximation algorithms for their related covering problems.

2.3.1 Bounded Degree and Frequency

We first discuss the most basic examples of all: bounded degree and bounded frequency. These are, respectively, set systems with few elements in each set, and set systems with few sets containing each element.

COVERING PROBLEM: Δ -**Regular-SC**
 ELEMENTS: Any finite set
 SETS: Finite subsets of X , each containing exactly Δ elements

In an instance of Δ -**Regular-SC**, each set is said to have *degree* Δ . We note that this forces a feasible solution to contain at least N/Δ sets. By a greedy method that repeatedly chooses the most cost-effective set remaining, it is not hard to obtain an algorithm for Δ -**Regular-SC** that attains an approximation ratio of

$$H_\Delta = 1 + \frac{1}{2} + \dots + \frac{1}{\Delta} \leq \log \Delta + 1,$$

where H_Δ is the Δ^{th} *harmonic number* [Vaz01]. Thus Δ -**Regular-SC** \in **APX** for any fixed value of Δ . For the value of $\Delta = 2$, this problem is equivalent to **Min-Edge-Cover** in graphs, which can be solved exactly in polynomial time via a method that employs a minimum-cost matching algorithm (such as that of Edmonds; see Chapter 27 of [Sch03]). However, for $\Delta \geq 3$, the problem has been shown to be **APX**-hard via a chain of L-reductions from **Min-Vertex-Cover** [AK00].

All of the above results can also be shown to hold when the definition of Δ -**Regular-SC** is modified so that each set contains *at most* Δ elements rather than *exactly* Δ elements. That is, the variation is still in **P** when $\Delta = 2$ and is still **APX**-complete when $\Delta \geq 3$.

We next discuss the ‘dual’ notion to bounded degree—bounded frequency:

2. FOUNDATIONS

COVERING PROBLEM: **k -Uniform-SC**

ELEMENTS: Any finite set

SETS: Finite subsets of X such that each element is contained in exactly k sets

In an instance of **k -Uniform-SC**, each element is said to have *frequency* k . It is not hard to show that simply selecting the cheapest set containing each element produces a k -approximation for weighted **k -Uniform-SC** [Vaz01]. However, the $k = 2$ case is precisely equivalent to **Min-Vertex-Cover**, which is **APX**-hard. Since the $k = 1$ case is trivial, it then follows that **k -Uniform-SC** is in **P** for $k = 1$ and is **APX**-complete for $k \geq 2$. Again, these results still hold when sets may contain fewer than k elements.

We also observe that **Δ -Regular-SC** and **k -Uniform-SC** are indeed *dual problems*: the dual of a k -uniform set system is a k -regular set system, and vice versa. The $k = 2$ case provides us with an interesting example—**Min-Vertex-Cover** is an **APX**-complete covering problem whose dual problem, **Min-Edge-Cover**, is polynomial-time solvable. Duality does not preserve **APX**-hardness or polynomial-time solvability.

Another noteworthy special case is the following:

COVERING PROBLEM: **3-Regular-Graph-SC**

ELEMENTS: The edges of a 3-regular graph $G = (V, E)$

SETS: $\{\delta(v) : v \in V\}$, where $\delta(v)$ is the set of edges incident to v

In **3-Regular-Graph-SC**, we have a combination of restrictions: sets have degree exactly 3 and the elements have frequency exactly 2. This covering problem is still **APX**-complete, as proven by Alimonti and Kann [AK00]. Reductions from **3-Regular-Graph-SC** are commonly used to prove other covering problems are **APX**-hard.

2.3.2 Hitting Set and Dominating Set

We have already discussed hitting set problems as the ‘dual’ of set cover problems in which the roles of sets and elements are reversed and the goal is to compute a min-cost subset of the elements so that each set contains at least one selected element (each set is ‘hit’ at least once). In the case of geometric problems, we will often try to describe such problems explicitly rather than via duality, because the resulting formulation carries physical intuition with it. For example, we can define the following:

COVERING PROBLEM: **\mathbb{R} -Interval-HS**

ELEMENTS: A finite set of intervals in \mathbb{R}

SETS: For a finite family of points P , the sets $\{S_p : p \in P\}$ where S_p comprises all of the intervals that contain p

Notation-wise, we will replace the ‘SC’ in a problem’s bolded name with ‘HS’ whenever we wish to examine the related hitting set dual problem. For example, we can examine **\mathbb{R}^2 -Rectangle-HS**, **Vertical-Tree-HS**, and so on. One easy result is the following:

Theorem 2.3.1. *Vertical-Tree-HS is in P.*

Proof. **Vertical-Tree-SC** is totally unimodular by Proposition 2.1.18, so its dual **Vertical-Tree-HS** is also by Proposition 2.1.15. The result then follows by Proposition 2.1.14. \square

We note that instead of trying to find a new combinatorial algorithm for the dual problem, we simply made use of the property of total unimodularity.

Dominating set problems are covering problems in which the elements and sets are one in the same:

COVERING PROBLEM: **Min-Dominating-Set**
 ELEMENTS: A finite family of sets \mathcal{S}
 SETS: The family $\{N(S) : S \in \mathcal{S}\}$, where $N(S) = \{A \in \mathcal{S} : A \cap S \neq \emptyset\}$

When given a set system (X, \mathcal{S}) , we may obtain the related *dominating set system* $(X, \mathcal{S})^D$ whose elements are the sets \mathcal{S} and whose sets are indexed by the sets \mathcal{S} , with an element being incident to a set if and only if the two corresponding sets intersect. In other words, the weighted **Min-Dominating-Set** problem asks us to select a minimum-cost family of sets $\mathcal{C} \subseteq \mathcal{S}$ such that every set in \mathcal{S} was either selected itself, or intersects a set that was selected. For a class \mathbf{C} of set systems, we shall write \mathbf{C}^D for the set $\{(X, \mathcal{S})^D : (X, \mathcal{S}) \in \mathbf{C}\}$ of related dominating set systems.

An equivalent way of viewing the *dominating set system* related to a set system (X, \mathcal{S}) is to consider the *intersection graph* $\Omega(\mathcal{S})$, which has \mathcal{S} as its vertex set and has vertices $S_i, S_j \in \mathcal{S}$ adjacent if and only if S_i and S_j have non-empty intersection.⁵ The dominating set problem is then simply a graph-theoretic covering problem in which the goal is to select a minimum cost subset of vertices in $\Omega(\mathcal{S})$ such that every unselected vertex is adjacent to at least one selected vertex. The set system matrix corresponding to an instance of **Min-Dominating-Set** is thus simply the adjacency matrix of the corresponding intersection graph (with ones added down the main diagonal, since each element covers itself). Since such matrices are symmetric, it follows that dominating set problems are *self-dual*:

Proposition 2.3.2. *Let (X, \mathcal{S}) be any set system. Then $(X, \mathcal{S})^{D*} = (X, \mathcal{S})^D$.*

As a consequence, it is unnecessary to discuss, e.g., hitting set variations of dominating set problems, as they are isomorphic to the original problem.

In the general case, it turns out that **Min-Dominating-Set** is equivalent to **Min-Set-Cover** under L-reductions [Kan92] and thus admits an $O(\log M)$ -approximation, but no better unless $\mathbf{P} = \mathbf{NP}$. Nevertheless, we will, as usual, examine several special cases that exhibit improved approximability.

One example is **R-Interval-DS**—the dominating set version of **R-Interval-SC**. In this problem, we are given a family of intervals on the real line, and we wish to select a minimum cost subfamily that intersects each interval in the input. As with **R-Interval-SC**, an approach based on dynamic programming yields an exact algorithm running in polynomial time, and thus:

⁵The theory of intersection graphs is a large area in and of itself; see [MM99].

Theorem 2.3.3. \mathbb{R} -*Interval-DS* is in \mathbf{P} .

We shall meet a few other dominating set problems in the chapters that follow.

2.3.3 Combining and Disassembling Set Systems

In this section, we cover a few ways by which individual set systems can be split apart, and multiple set systems can be merged.

Definition 2.3.4. Given two set system matrices A_1 and A_2 both having the same number of rows, we write $A_1 \parallel A_2$ for the *concatenation* of A_2 onto the right of A_1 .

Concatenation produces a set system $(X, \mathcal{S}_1 \cup \mathcal{S}_2)$ from two set systems (X, \mathcal{S}_1) and (X, \mathcal{S}_2) having the same ground set. This allows us to represent, for example, geometric set systems in which the goal is to cover points in the plane using both disks and rectangles.

We shall also define concatenation over entire covering problems:

Definition 2.3.5. Given two classes \mathbf{C}_1 and \mathbf{C}_2 of set systems, we write $\mathbf{C}_1 \parallel \mathbf{C}_2$ for the class of all set systems of the form $A_1 \parallel A_2$ with $A_1 \in \mathbf{C}_1$, $A_2 \in \mathbf{C}_2$, and both A_1 and A_2 having the same number of rows.

As an example, consider the following special case of \mathbb{R}^2 -**Rectangle-SC**:

COVERING PROBLEM: \mathbb{R}^2 -**Slab-SC**
 ELEMENTS: A finite subset of \mathbb{R}^2
 SETS: A family of axis-aligned slab-shaped rectangular regions, each either of the form $[a, b] \times [-\infty, \infty]$ or of the form $[-\infty, \infty] \times [a, b]$

If we take an \mathbb{R}^2 -**Slab-SC** matrix A and write it as $A_1 \parallel A_2$ where the columns in A_1 only correspond to horizontal slabs and the columns of A_2 only correspond to vertical slabs, then both A_1 and A_2 are \mathbb{R} -**Interval-SC** matrices. The following shall provide a method with which we can use this structural property to obtain a constant approximation for \mathbb{R}^2 -**Slab-SC**:

Theorem 2.3.6. *Suppose \mathbf{C}_1 and \mathbf{C}_2 are two hereditary families of set systems that both admit an LP-relative polynomial-time α -approximation for their unweighted (respectively, weighted) covering problems for some $\alpha \geq 1$. Then the unweighted (respectively, weighted) covering problem on $\mathbf{C}_1 \parallel \mathbf{C}_2$ admits an LP-relative polynomial-time (2α) -approximation.*

Proof. We prove the weighted version; the unweighted version is similar. Let $A_1 \in \mathbf{C}_1$ be $M \times N_1$ and let $A_2 \in \mathbf{C}_2$ be $M \times N_2$, and let $\mathbf{w}_1 \in \mathbb{R}_+^{N_1}$ and $\mathbf{w}_2 \in \mathbb{R}_+^{N_2}$ be two sets of weights. We shall provide an algorithm to approximate the optimal solution to the weighted covering problem on $A_1 \parallel A_2$ with weights $\mathbf{w}_1 \parallel \mathbf{w}_2$.

We first solve the linear programming relaxation (SCLP) of the covering problem involving the set system $A_1 \parallel A_2$ and weights $\mathbf{w}_1 \parallel \mathbf{w}_2$. Let \mathbf{x}^* be an

optimal fractional solution and define $\mathbf{x}' = 2\mathbf{x}^*$. Write \mathbf{x}' as $\mathbf{x}'_1 \parallel \mathbf{x}'_2$ for the respective fractional weights of columns in A_1 and A_2 . Since $A\mathbf{x}^* \geq \mathbf{1}$, each row of $A\mathbf{x}'$ is at least 2 and thus for all i , we must have either $[A_1\mathbf{x}'_1]_i \geq 1$ or $[A_2\mathbf{x}'_2]_i \geq 1$. Mark row i of $A_1 \parallel A_2$ if $[A_1\mathbf{x}'_1]_i \geq 1$, and leave the remaining rows unmarked. Let A'_1 be the submatrix of A_1 containing only marked rows, and let A'_2 be the submatrix of A_2 containing only unmarked rows.

By our marking scheme, $A'_1\mathbf{x}'_1 \geq \mathbf{1}$ and $A'_2\mathbf{x}'_2 \geq \mathbf{1}$, so \mathbf{x}'_1 and \mathbf{x}'_2 are feasible solutions of the covering problems on A'_1 and A'_2 respectively. Since \mathbf{C}_1 and \mathbf{C}_2 are hereditary, we have $A'_1 \in \mathbf{C}_1$ and $A'_2 \in \mathbf{C}_2$ and thus the two weighted covering problems on A'_1 (with weights \mathbf{w}_1) and A'_2 (with weights \mathbf{w}_2) both admit LP-relative polynomial-time α -approximations. Let \mathbf{x}_1^\dagger and \mathbf{x}_2^\dagger be α -approximate integer solutions to these two covering problems obtained via such an algorithm. Observe that $\mathbf{x}_1^\dagger \parallel \mathbf{x}_2^\dagger$ is a feasible integer solution for our original covering problem on $A_1 \parallel A_2$, since both marked and unmarked rows are covered. It remains to bound the cost of this solution, proving our approximation guarantee:

$$\begin{aligned} (\mathbf{w}_1 \parallel \mathbf{w}_2)^T(\mathbf{x}_1^\dagger \parallel \mathbf{x}_2^\dagger) &= \mathbf{w}_1^T \mathbf{x}_1^\dagger + \mathbf{w}_2^T \mathbf{x}_2^\dagger \\ &\leq \alpha(\mathbf{w}_1^T \mathbf{x}'_1 + \mathbf{w}_2^T \mathbf{x}'_2) \\ &\leq 2\alpha(\mathbf{w}_1^T \mathbf{x}_1^* + \mathbf{w}_2^T \mathbf{x}_2^*) \\ &= 2\alpha(OPT) \end{aligned}$$

This shows that our algorithm does indeed yield an LP-relative polynomial-time (2α) -approximation, completing the proof. \square

We may apply Theorem 2.3.6 to \mathbb{R}^2 -**Slab-SC** with $\alpha = 1$ since \mathbb{R} -**Interval-SC** matrices are TUM, yielding the following:

Corollary 2.3.7. *Weighted \mathbb{R}^2 -**Slab-SC** admits a polynomial-time LP-relative 2-approximation.*

A related, but entirely separate idea is that of splitting the sets of an individual set system into multiple smaller sets, and solving the resulting problem. We have already encountered one example of this idea in Theorem 1.2.4, where we obtained a 2-approximation for **Tree-SC** by separating each path in a tree into two vertical paths. The following is a more general notion:

Definition 2.3.8. Let A_1 and A_2 be two set system matrices, both of dimension $M \times N$. The matrix $A_1 \vee A_2$ is the $M \times N$ matrix obtained by performing a ‘element-wise OR operation’ on A_1 and A_2 . In other words,

$$[A_1 \vee A_2]_{i,j} = \begin{cases} 1 & : [A_1]_{i,j} = 1 \text{ or } [A_2]_{i,j} = 1 \\ 0 & : \text{otherwise} \end{cases}$$

We shall also define element-wise operations over entire covering problems:

Definition 2.3.9. Given two classes \mathbf{C}_1 and \mathbf{C}_2 of set systems, we write $\mathbf{C}_1 \vee \mathbf{C}_2$ for the class of all set systems of the form $A_1 \vee A_2$ with $A_1 \in \mathbf{C}_1$, $A_2 \in \mathbf{C}_2$, and both A_1 and A_2 having the same dimensions.

2. FOUNDATIONS

Via precisely the same mechanism as the one used to prove Theorem 1.2.4, we may obtain the following more general result:

Theorem 2.3.10. *Suppose \mathbf{C} and \mathbf{D} are two families of set systems such that for all $A \in \mathbf{C}$, we may find, in polynomial time, matrices B_1 and B_2 such that $A = B_1 \vee B_2$ and $B_1 \parallel B_2 \in \mathbf{D}$. Suppose also that the unweighted (respectively, weighted) covering problem on \mathbf{D} admits a polynomial-time α -approximation. Then the unweighted (respectively, weighted) covering problem on \mathbf{C} admits a polynomial-time (2α) -approximation.*

Proof. Similar to the proof of Theorem 1.2.4. If we wish to solve the covering problem on A with weights \mathbf{w} , we first find an α -approximate solution for the covering problem on $B_1 \parallel B_2$ (with weights inherited), and then take the corresponding columns in A . This yields a 2α -approximate solution since a min-cost solution for A must cost at least half as much as a min-cost solution for $B_1 \parallel B_2$. \square

Remark 2.3.11. Theorem 2.3.10 will produce an LP-relative approximation if the α -approximation for \mathbf{D} is LP-relative.

This can be applied to the hitting set version of \mathbb{R}^2 -**Slab-HS**:

Corollary 2.3.12. *Weighted \mathbb{R}^2 -**Slab-HS** admits a polynomial-time LP-relative 2-approximation.*

Proof. Given a \mathbb{R}^2 -**Slab-HS** matrix A (that is, the transpose of a \mathbb{R}^2 -**Slab-SC** matrix), let B_1 be A with all rows corresponding to horizontal slabs replaced by rows of zeroes, and let B_2 be A with all rows corresponding to vertical slabs replaced by rows of zeroes. Clearly $A = B_1 \vee B_2$, and $B_1 \parallel B_2$ is a \mathbb{R} -**Interval-HS** matrix (one can think of it as representing two entirely independent \mathbb{R} -**Interval-HS** problems in the same matrix), so the covering problem on $B_1 \parallel B_2$ is TUM and has an integrality gap of 1. The result then follows by Theorem 2.3.10. \square

Remark 2.3.13. Theorems 2.3.6 and 2.3.10 can both easily be generalized to the concatenation or element-wise OR of more than 2 matrices, with appropriate worsening in the approximation guarantee.

At this point, we have shown that both \mathbb{R}^2 -**Slab-SC** and \mathbb{R}^2 -**Slab-HS** instances can be created by elementary constructions involving very simple TUM matrices, and that these constructions facilitate constant approximations. In fact, by combining Theorems 2.3.6 and 2.3.10, constant approximations can be obtained for *any* class of matrices that admit a polynomial-time decomposition into the element-wise OR of a bounded number of TUM matrices or other matrices admitting an LP-relative constant approximation. Below is a final example that uses several of the tools we have developed:

Theorem 2.3.14. *Weighted **Tree-HS** admits a polynomial-time LP-relative 4-approximation.*

Proof. Let \mathbf{T} be the class of TUM matrices. By Theorem 2.3.6, since \mathbf{T} is hereditary, weighted covering on $\mathbf{T} \parallel \mathbf{T}$ admits a polynomial-time LP-relative 2-approximation. Consequently, weighted covering on $\mathbf{T} \vee \mathbf{T}$ admits a polynomial-time LP-relative 4-approximation provided a decomposition into the element-wise OR of two TUM matrices can be found in polynomial time. We shall show that this is the case for **Tree-HS** matrices.

Let A be a **Tree-HS** matrix. Since **Tree-SC** matrices can be decomposed into the element-wise OR of two **Vertical-Tree-SC** matrices, write $A^T = B_1 \vee B_2$ for two **Vertical-Tree-SC** matrices B_1 and B_2 . Note that the transpose operation distributes over element-wise OR, and thus it follows that $A = B_1^T \vee B_2^T$. The **Vertical-Tree-SC** matrices B_1 and B_2 are TUM by Theorem 2.1.18, so $B_1^T \in \mathbf{T}$ and $B_2^T \in \mathbf{T}$ by Proposition 2.1.15 and hence $A \in \mathbf{T} \vee \mathbf{T}$. The result follows. \square

Given how straightforward these types of decompositions are, one might hope that something better, such as a PTAS or polynomial-time solvability, might be possible, especially for \mathbb{R}^2 -**Slab-SC** or \mathbb{R}^2 -**Slab-HS**. Unfortunately, we will prove in Chapter 5 that \mathbb{R}^2 -**Slab-SC** and \mathbb{R}^2 -**Slab-HS** are both **APX**-hard; the concatenation or element-wise OR of two polynomial-time solvable covering problems may not even admit a PTAS.

2.3.4 Geometric Set Systems

Computational geometry is a rich source of set systems having nice structural properties. In a *geometric set cover problem*, the elements X are points in Euclidean space, and the sets \mathcal{S} are a pre-specified configuration of regions or geometric objects. We have already seen examples like \mathbb{R}^2 -**Rectangle-SC**; here we discuss some more exotic set systems and prove some useful properties about them. Many of these properties can be found in a standard textbook such as [dBCvKO08].

We begin with a class of set systems relevant to wireless network planning [GRV05]:

COVERING PROBLEM: \mathbb{R}^n -**Ball-SC**
 ELEMENTS: A finite subset of \mathbb{R}^n
 SETS: A family of closed balls, each of the form $B_r[\mathbf{p}]$ for some $\mathbf{p} \in \mathbb{R}^n$ and some $r \geq 0$ (where $B_r[\mathbf{p}]$ is the set containing all points whose Euclidean distance to \mathbf{p} is at most r)

A variation of this is \mathbb{R}^n -**Unit-Ball-SC**, in which every ball has radius 1. In the plane, these problems will be referred to by their traditional names \mathbb{R}^2 -**Disk-SC** and \mathbb{R}^2 -**Unit-Disk-SC**. It turns out that the unit radius restriction can be enforced by adding an additional Euclidean dimension:

Proposition 2.3.15. \mathbb{R}^n -**Ball-SC** \subseteq \mathbb{R}^{n+1} -**Unit-Ball-SC**.

Proof. Let (X, \mathbf{S}) be an \mathbb{R}^n -**Ball-SC** instance. Without loss of generality, we assume that all balls have radius at most 1. We employ the following mappings, which send points and balls in \mathbb{R}^n to points and unit balls in \mathbb{R}^{n+1} :

2. FOUNDATIONS

- $(x_1, \dots, x_n) \mapsto (x_1, \dots, x_n, 0)$
- $B_r[(x_1, \dots, x_n)] \mapsto B_1[(x_1, \dots, x_n, \sqrt{1-r^2})]$

We can easily check that

$$(x_1, \dots, x_n) \in B_r[(x_1, \dots, x_n)] \iff (x_1, \dots, x_n, 0) \in B_1[(x_1, \dots, x_n, \sqrt{1-r^2})],$$

implying that the above mapping sends \mathbb{R}^n -**Ball-SC** instances to isomorphic \mathbb{R}^{n+1} -**Unit-Ball-SC** instances. The result follows. \square

The above style of proof shall become a very common motif—we will often verify encoding relations by explicitly defining a geometric mapping that preserves the element-set incidence relation of desired set systems.

It turns out that \mathbb{R}^n -**Unit-Ball-SC** is a *self-dual* problem; its hitting set version can encode precisely the same problems as its covering version:

Proposition 2.3.16. \mathbb{R}^n -**Unit-Ball-SC** = \mathbb{R}^n -**Unit-Ball-HS**

Proof. Since $\mathbf{x} \in B_1[\mathbf{y}]$ if and only if $\mathbf{y} \in B_1[\mathbf{x}]$, it suffices to simply replace each point \mathbf{x} with a unit ball centred at \mathbf{x} , and each unit ball centered at \mathbf{y} with a point \mathbf{y} . \square

For the same reason, \mathbb{R}^2 -**Unit-Square-SC** is also self-dual.

In a configuration of disks in the plane, no pair of disks have boundaries that intersect more than twice. This topological property is so useful that many authors have studied the following generalization of \mathbb{R}^2 -**Disk-SC**:

COVERING PROBLEM: \mathbb{R}^2 -**Pseudodisk-SC**
 ELEMENTS: A finite subset of \mathbb{R}^2
 SETS: A family of connected, closed, bounded regions, each of which has a simple closed Jordan curve as its boundary, with each pair of Jordan curves intersecting at most twice

\mathbb{R}^2 -**Pseudodisk-SC** encodes \mathbb{R}^2 -**Disk-SC** as well as various other problems such as \mathbb{R}^2 -**Square-SC**. However, it cannot encode \mathbb{R}^2 -**Rectangle-SC**, as it is possible for the boundaries of two axis-aligned rectangles to intersect four times.

Our next example is covering involving *lower half-spaces*. For our purposes, a half-space in \mathbb{R}^n is a lower half-space whenever its intersection with the negative x_n -coordinate axis is a ray of infinite length.

COVERING PROBLEM: \mathbb{R}^n -**Lower-Halfspace-SC**
 ELEMENTS: A finite subset of \mathbb{R}^n
 SETS: A family of closed lower half-spaces in \mathbb{R}^n

It turns out that \mathbb{R}^n -**Lower-Halfspace-SC** can also encode \mathbb{R}^n -**Ball-SC**. Like in Proposition 2.3.15, the transformation requires adding another dimension:

Proposition 2.3.17. \mathbb{R}^n -**Ball-SC** \subseteq \mathbb{R}^{n+1} -**Lower-Halfspace-SC**.

Proof. This can be shown via a mapping known as the *standard lifting transformation*; see [dBCvKO08] for details. As an explicit example, for the case of $n = 2$, it suffices to employ the following mappings, which send points and balls in the plane to points and lower half-spaces in \mathbb{R}^3 , preserving incidence relations:

- $(a, b) \mapsto (a, b, a^2 + b^2)$
- $B_r[a, b] \mapsto \{(x, y, z) : z - 2ax - 2by + a^2 + b^2 - r^2 \leq 0\}$

Analogous mappings can be used in higher dimensions. □

Using tools of projective geometry, we can show that \mathbb{R}^n -**Lower-Halfspace-SC** is also a self-dual problem:

Proposition 2.3.18. \mathbb{R}^n -**Lower-Halfspace-HS** = \mathbb{R}^n -**Lower-Halfspace-SC**

Proof. Assuming no points or half-space boundaries intersect the origin, it is sufficient to apply a *projective duality* mapping, which maps points to lower half-spaces and vice-versa, preserving incidence properties. See [dBCvKO08] for details. As an explicit example, for the case of $n = 2$, we may send points (a, b) to lower half-planes $ax + by + 1 \leq 0$ and vice versa. □

Next, we define a broad class of covering problems known collectively as *box cover* problems. The set systems involved in these covering problems have numerous applications in areas such as database range-searching and computational learning theory [HW87].

COVERING PROBLEM: \mathbb{R}^n -($n + k$)-Sided-Box-SC
ELEMENTS: A finite subset of \mathbb{R}^n
SETS: A family of $(n + k)$ -sided axis-aligned boxes, each of the form $[a_1, b_1] \times \dots \times [a_k, b_k] \times [-\infty, b_{k+1}] \times \dots \times [-\infty, b_n]$ where all $a_i, b_i \in \mathbb{R}$ and $a_i \leq b_i$ for $1 \leq i \leq k$

Note that we require $0 \leq k \leq n$ in the above formulation. In the event that $k = n$, this problem is simply geometric set cover with ordinary n -dimensional axis-aligned boxes. For example, \mathbb{R}^1 -**2-Sided-Box-SC** is just \mathbb{R} -**Interval-SC**, and \mathbb{R}^2 -**4-Sided-Box-SC** is simply \mathbb{R}^2 -**Rectangle-SC**. The crucial part of the problem definition is that in the event that $k < n$, each box is always semi-infinite in the negative direction in the first k dimensions. This allows us to describe many new types of covering problems. For example, in \mathbb{R}^2 -**2-Sided-Box-SC**, the goal is to cover points in the plane using quadrants, each of which extends semi-infinitely downward and leftward in the plane.

When \mathbb{R}^n -($n + k$)-**Sided-Box-SC** instances are considered in the literature, the boxes are sometimes regarded as having fixed endpoints in some dimensions, instead of extending off to infinity. For example, \mathbb{R}^2 -**3-Sided-Box-SC** is equivalent to the problem of covering with axis-aligned rectangles, each of which abuts the x -axis.⁶ These two representations are isomorphic in that any

⁶For this reason, \mathbb{R}^2 -**3-Sided-Box-SC** has been called *hinged axis-aligned rectangle cover* or HARC [BKS11].

2. FOUNDATIONS

instance of one representation can easily be reformulated using the other, and the reader is encouraged to pick whichever visualization is most convenient. We prefer semi-infinite boxes because in such a case, many properties of these problems (such as their VC dimension, see Section 2.4) are a function of the number of sides of the boxes.

The complexity of a \mathbb{R}^n - $(n+k)$ -**Sided-Box-SC** problem is a monotonically increasing function of both the number of sides and the number of dimensions:

Proposition 2.3.19. *For $0 \leq k < n$, the following always hold:*

1. \mathbb{R}^n - $(n+k)$ -**Sided-Box-SC** \subseteq \mathbb{R}^n - $(n+k+1)$ -**Sided-Box-SC**.
2. \mathbb{R}^{n-1} - $(n+k)$ -**Sided-Box-SC** \subseteq \mathbb{R}^n - $(n+k)$ -**Sided-Box-SC**.

Proof. Item (1) is obvious. To verify (2), we give only a proof that \mathbb{R} -**Interval-SC** = \mathbb{R}^1 -**2-Sided-Box-SC** \subseteq \mathbb{R}^2 -**2-Sided-Box-SC**; all other cases can be checked analogously.

Given an instance of \mathbb{R} -**Interval-SC**, consider the following mapping sending real numbers and intervals on the real line to points and quadrants in the plane:

- $x \mapsto (x, -x)$
- $[a, b] \mapsto [-\infty, b] \times [-\infty, -a]$

We can easily check that $x \in [a, b]$ if and only if $(x, -x) \in [-\infty, b] \times [-\infty, -a]$, implying that the above mapping sends \mathbb{R} -**Interval-SC** instances to isomorphic \mathbb{R}^1 -**2-Sided-Box-SC** instances. The result follows. \square

Box cover problems in which *all* dimensions are semi-infinite are self-dual:

Proposition 2.3.20. \mathbb{R}^n - n -**Sided-Box-SC** = \mathbb{R}^n - n -**Sided-Box-HS**

Proof. We give a proof for $n = 2$; other cases are similar. It is straightforward to check that the following bijective mapping sending points to quadrants and vice-versa preserves the necessary incidence relations in both directions:

- $(x, y) \mapsto [-\infty, -x] \times [-\infty, -y]$
- $[-\infty, x] \times [-\infty, y] \mapsto (-x, -y)$

\square

Somewhat surprisingly, the problem \mathbb{R}^2 -**3-Sided-Box-SC** can encode the problem **Vertical-Tree-SC**. In a sense, this implies that the geometric \mathbb{R}^2 -**3-Sided-Box-SC** problem is a stronger generalization of \mathbb{R} -**Interval-SC** than the combinatorial **Vertical-Tree-SC** problem:

Theorem 2.3.21. **Vertical-Tree-SC** \subseteq \mathbb{R}^2 -**3-Sided-Box-SC**

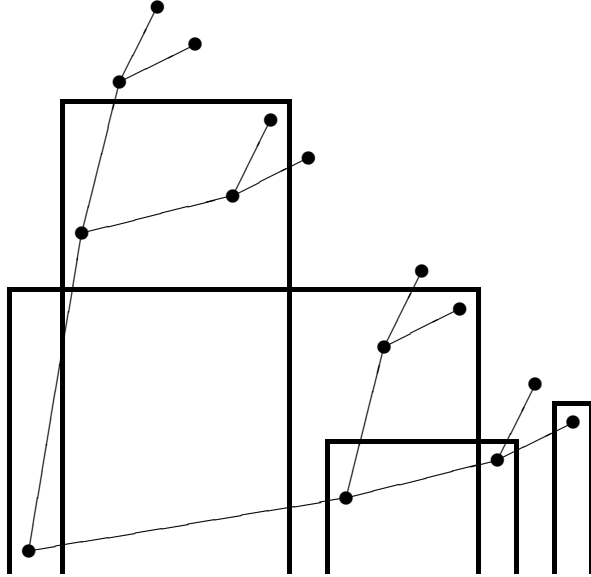


Figure 2.1: Encoding of a **Vertical-Tree-SC** instance as a \mathbb{R}^2 -**3-Sided-Box-SC** instance. Each point represents an *edge* from a full binary tree T . The thin lines are for visualization purposes only, and connect points that share a parent-child bond in T . The axis-aligned 3-sided boxes each cover a set of points corresponding to a vertical path in T .

Proof. We have given a rigorous proof in [CGK10b], but we shall give a simpler ‘proof by picture’ that, we believe, provides a much clearer indication of precisely how the structure of a **Vertical-Tree-SC** instance can be exploited to obtain a \mathbb{R}^2 -**3-Sided-Box-SC** instance.

Let (X, \mathcal{S}) be a **Vertical-Tree-SC** instance on a tree T . Without loss of generality, we may assume that T is a subgraph of a full binary rooted tree (by replacing each high-degree vertex in T with a binary tree, adjusting the paths accordingly, and deleting unnecessary edges). We then create a point in the plane for each edge in T as shown in Figure 2.1, and observe that the edges covered by any vertical path in T can be covered by a single 3-sided box that contains no other points. This completes the proof. \square

We shall discuss several other geometric set systems involving convex objects such as polygons in the plane. However, additional restrictions are necessary, since geometric covering and hitting set problems with arbitrary convex polygons can encode general **Min-Set-Cover**. For example, by placing a series of points on a circle, it is easy to see that any subset of them can be contained in a convex polygon that contains no other points on the circle. Even restricting the polygons to triangles is sufficient to encode **3-Regular-Graph-SC**, implying that \mathbb{R}^2 -**Triangle-Cover** is **APX-hard**.

With hitting set problems, there are similar issues with other types of geometric set systems. Hitting set with arbitrary line segments in the plane can

2. FOUNDATIONS

easily encode **Min-Vertex-Cover**, again by simply placing points on a circle. It follows that, for example, **Min-Vertex-Cover** \subseteq **\mathbb{R}^2 -Triangle-Hitting-Set** and thus **\mathbb{R}^2 -Triangle-Hitting-Set** is **APX**-hard (since any line segment can be replaced by an appropriate skinny triangle).

To circumvent the difficulties associated with skinny geometric objects, some authors have concentrated on restricted geometric set systems in which each object must be sufficiently ‘fat’. There are various definitions of ‘fatness’ used in the literature, most of which are more-or-less equivalent [Cha03, EAS11]. We will use the following definition:

Definition 2.3.22. A bounded, convex⁷ geometric object S in the plane is α -fat if there exist circles C_1 and C_2 with radii r_1 and r_2 such that $C_1 \subseteq S \subseteq C_2$ and $\frac{r_2}{r_1} < \alpha$.

For example, disks are 1-fat and squares are $\sqrt{2}$ -fat. We shall discover, in general, that problems involving fat objects often admit better approximations than those with skinny objects.

2.3.5 Priority Set Systems

In a *priority covering problem*, we are given a set system (X, \mathcal{S}) , a priority supply vector $\mathbf{s} \in \mathbb{Z}_+^N$, a priority demand vector $\pi \in \mathbb{Z}_+^M$, and (in the weighted case) weights $\mathbf{w} \in \mathbb{R}_+^N$. The goal is to select a minimum-weight subset $\mathcal{C} \subseteq \mathcal{S}$ such that for each $x \in X$, there is a selected set $S \in \mathcal{C}$ such that $x \in S$ and $s_S \geq \pi_x$. In other words, we wish to purchase a subset of \mathcal{S} so that each point x is covered by a set whose supply exceeds the demand of x .

Priority covering problems arise naturally in situations where *quality of service* restrictions are important—we may think of each set as providing a certain level of service to all of the elements it contains, each element being satisfied only if its required level of service is met. Priority versions of the Steiner tree and Steiner forest problems were first studied by Charikar, Naor and Schieber [CNS04], motivated by network multicast routing problems in which varying qualities of data are simultaneously broadcasted. We shall study the priority versions of problems like **Tree-SC** by relating them to geometric covering problems.

We can express the priority relationship via a matrix:

Definition 2.3.23. Given priority supplies and demands $\mathbf{s} \in \mathbb{Z}_+^N$ and $\pi \in \mathbb{Z}_+^M$, the *priority matrix* $\Pi[\mathbf{s}, \pi]$ is an $M \times N$ binary matrix filled as follows:

$$\Pi[\mathbf{s}, \pi]_{ij} = \begin{cases} 1 & : s_j \geq \pi_i \\ 0 & : \text{otherwise,} \end{cases}$$

If the rows and columns of $\Pi[\mathbf{s}, \pi]$ are permuted to appear according to a non-decreasing order of their priorities (from left to right and bottom to top

⁷For non-convex objects, there are more general and sophisticated measures of fatness such as *local fatness* that preserve the useful combinatorial properties of fat objects [dB10]

respectively), then $\Pi[\mathbf{s}, \pi]$ takes on a *staircase* structure in which no one appears immediately above or left of any zero. It follows that that priority matrices are a special case of \mathbb{R} -**Interval-SC** matrices and are thus TUM. Moreover, this staircase structure is preserved when transposing the matrix, so the transpose of a priority matrix is a priority matrix.

When we add priorities to a set system, the resulting *priority set system* is simply another set system in which some element-set incidences are removed (some elements may no longer be covered by sets containing them if the set's supply is not high enough).

Definition 2.3.24. Given a set system matrix A and priority supplies and demands $\mathbf{s} \in \mathbb{Z}_+^N$ and $\pi \in \mathbb{Z}_+^M$, we define the *priority set system matrix* $A[\mathbf{s}, \pi]$ to be $A \wedge \Pi[\mathbf{s}, \pi]$, where \wedge is the 'element-wise AND' operation on matrices. In other words, $A[\mathbf{s}, \pi]_{ij} = 1$ if and only if $A_{ij} = 1$ and $s_j \geq \pi_i$.

We shall also define a class of set systems for *all* of the priority covering instances related to a given class of set systems:

Definition 2.3.25. For a class of set systems \mathbf{C} , define \mathbf{C}^P to be the set of all priority set system matrices $A \wedge \Pi$ over all choices of $A \in \mathbf{C}$ and all appropriately sized priority matrices Π .

An example of a priority covering problem that we will encounter multiple times is **Priority- \mathbb{R} -Interval-SC**—the priority version of \mathbb{R} -**Interval-SC**. A geometric view of priorities shall help us visualize this problem more easily. Imagine a **Priority- \mathbb{R} -Interval-SC** instance (X, \mathcal{S}) where each point $x \in \mathbb{R}$ is placed in the plane at coordinates (x, π_x) and each interval $S = [a, b] \subseteq \mathbb{R}$ is replaced with a line segment connecting (a, s_S) to (b, s_S) . In other words, we have taken our 1-dimensional covering problem and extruded it out into the second dimension, with the final coordinate of each object representing its priority (supply or demand). In this representation, an interval S represented by a segment connecting (a, s_S) to (b, s_S) covers a point x represented by the point (x, π_x) if and only if $a \leq x \leq b$ and $\pi_x \leq s_S$, which happens precisely when (x, π_x) lies in the *downward shadow* of the segment.

Definition 2.3.26. For a set Y of points in the plane, we define the *downward shadow* of Y to be the set of all points (a, b) such that there is a point $(a, y) \in Y$ with $y \geq b$.

However, the downward shadow of a horizontal line segment in the plane is precisely a 3-sided box. This leads us to the following observation:

Proposition 2.3.27. *Priority- \mathbb{R} -Interval-SC is isomorphic to \mathbb{R}^2 -3-Sided-Box-SC.*

3-sided boxes are sometimes called *bottomless rectangles* [Kes07], and covering problems involving them are fairly well-studied and arise in a variety of

2. FOUNDATIONS

situations [BP10, BKS11]. In [CGK10a], we showed that **Priority- \mathbb{R} -Interval-SC** can be solved exactly in polynomial time via a dynamic programming algorithm. We give a simpler and more general proof of this in Chapter 4 (see also [CG11]).

Intuitively, for a geometric covering problem, we can think of the process of adding priorities as that of adding another Euclidean dimension in which the *height* of each point or object is determined by its height, and an object covers precisely the points in its downward shadow. For box cover problems, the priority version of a problem is obtained by adding another dimension in which all objects are semi-infinite:

Proposition 2.3.28. \mathbb{R}^n - $(n+k)$ -**Sided-Box-SC**^P = \mathbb{R}^{n+1} - $(n+k+1)$ -**Sided-Box-SC**

Some other basic results about the priority operation follow:

Proposition 2.3.29. *Let \mathbf{C} and \mathbf{D} be classes of set systems. Then the following are true:*

1. *If \mathbf{C} is hereditary, then so is \mathbf{C}^P .*
2. *If $\mathbf{C} \subseteq \mathbf{D}$, then $\mathbf{C}^P \subseteq \mathbf{D}^P$.*
3. *$(\mathbf{C}^*)^P = (\mathbf{C}^P)^*$.*

Proof. The first two items are immediate from the definitions. The third is clear through the following chain of implications:

$$\begin{aligned}
 B \in (\mathbf{C}^*)^P &\iff B = A^T \wedge \Pi \text{ for some } A \in \mathbf{C} \text{ and priority matrix } \Pi \\
 &\iff B = (A \wedge \Pi^T)^T \text{ for some } A \in \mathbf{C} \text{ and priority matrix } \Pi \\
 &\iff B = (A \wedge \Pi)^T \text{ for some } A \in \mathbf{C} \text{ and priority matrix } \Pi \\
 &\iff B \in (\mathbf{C}^P)^*.
 \end{aligned}$$

Note that we used the fact that transposes of priority matrices are priority matrices, and the fact that matrix transpose distributes over element-wise AND. \square

2.4 Measuring and Exploiting the Complexity of Set Systems

If we are to obtain general algorithms that achieve better approximations for covering problems by exploiting their structural properties, it would be useful to first have a quantifiable way of identifying structure. It might first appear tempting to use global properties of a set system, such as the average number of elements per set, as an indicator of set system complexity. However, large set systems that appear simple and unimposing at first glance may conceal

difficult subproblems within them, and global properties may fail to detect these pathological subproblems. Consequently, more involved methods that analyze a matrix locally are required.

The approach used traditionally in practice can be stated as follows: first, establish some canonical, infinite list $\{A_0, A_1, \dots\}$ of matrices that are nondecreasing in size and complexity as their index increases. Then, characterize a class of set systems \mathbf{C} according to the largest value of d for which A_d is a submatrix of some $A \in \mathbf{C}$. If $\{A_0, A_1, \dots\}$ is chosen well, then ruling out all submatrices $\{A_{d+1}, A_{d+2}, \dots\}$ shall force severe restrictions upon the structure of A . For some choices of forbidden submatrices, these restrictions can yield necessary or sufficient conditions for obtaining an improved approximation.

The most important set system complexity measures that will be discussed in this thesis are:

- *Vapnik-Chervonenkis dimension* or more commonly, VC dimension—a metric originally developed in computational learning theory that can be exploited to obtain improved approximability for covering problems; indeed, to admit *any* improved approximability, a hereditary class of set systems must exhibit bounded VC dimension.
- *Union Complexity*—used primarily in computational geometry to quantify the complexity of the boundary of a union of geometric objects; set systems exhibiting low union complexity admit improved LP-relative approximations.
- *Shallow Cell Complexity*—a new notion that we originally introduced in [CGKS12]; in Chapter 6 we provide an algorithm for weighted set cover whose factor of approximation is a function of the shallow cell complexity of the set system. Many common set systems, including those of low union complexity, exhibit low shallow cell complexity.

2.4.1 VC dimension

VC dimension was first introduced by Vapnik and Chervonenkis in 1971 in a groundbreaking paper that led to decades of development in the areas of computational and statistical learning theory [VC71, BEHW89]. In short, VC dimension is the single key property of a set system that determines how quickly a computational process can learn to statistically classify points, and many statistical properties of machine learning algorithms are related to the VC dimension of the underlying set system. A rich theory has developed surrounding these ideas; for more information, one may refer to any one of a number of texts on the subject such as Vapnik’s [Vap98]. We shall only make use of a select few key ideas from this theory. For our purposes, it shall suffice to treat the VC dimension of a set system as a forbidden submatrix property.

Definition 2.4.1. For an integer $d \geq 1$, define the matrix V_d to be the $d \times 2^d$ set system matrix whose columns each contain a distinct d -bit binary string (of the 2^d such strings).

2. FOUNDATIONS

Recalling that we consider set system matrices *modulo row and column reorderings*, we remark that the matrix V_d satisfying the above property is unique (and thus well-defined) and contains no two identical rows or columns. The matrix V_d is the largest possible set system matrix containing d rows, since adding more columns to V_d would result in some column being duplicated.

Definition 2.4.2. The *VC dimension* of a matrix A , written $\text{VC}(A)$, is the largest value of d such V_d is a submatrix of A (where, as usual, the submatrix relation is modulo row and column reorderings).

More traditionally, if a set system (X, \mathcal{S}) admits V_d as a submatrix of its set system matrix, then there must exist a d -element subset $D \subseteq X$ such that there is some subfamily $\mathcal{T} \subseteq \mathcal{S}$ containing 2^d sets, each of which is distinct when restricted to D . In such a case, \mathcal{T} is said to *shatter* D , and the VC dimension of a set system is simply the size of the largest set that can be shattered.

It is known that the VC dimension of a given, explicitly represented set system matrix can be computed in time $O(n^{\log n})$ [LMR91], but computing it exactly is hard for the class **LOGNP** [PY96]. When a set system matrix is not given, and instead a polynomial-size circuit is provided that answers element-set incidence queries for a potentially exponential-sized set system, computing the VC dimension becomes **NP-hard**; in fact, under this model, it is **NP-hard** to approximate VC dimension to within a factor of $2 - \epsilon$ for any $\epsilon > 0$ [MU02].⁸

We note that all non-trivial set system matrices have a VC dimension of at least 1. In general, simpler set systems have lower VC dimension. Accordingly, we define the VC dimension of a class of set systems by the worst case VC dimension among its constituent matrices.

Definition 2.4.3. Given a class \mathbf{C} of set system matrices, its *VC dimension* $\text{VC}(\mathbf{C})$ is the maximum of $\text{VC}(A)$ over all $A \in \mathbf{C}$.

A simpler definition is possible for hereditary classes of set systems. The following is immediate from the previous definitions and the definition of heredity:

Proposition 2.4.4. *If \mathbf{C} is hereditary, then $\text{VC}(\mathbf{C})$ is the largest value of d for which $V_d \in \mathbf{C}$.*

VC dimension also has a dual notion:

Definition 2.4.5. The *VC codimension* $\text{VC}^*(A)$ of a matrix A is the largest value of d such that V_d^T is a submatrix of A . Equivalently, it is the VC dimension of A^T . For a class of set systems \mathbf{C} , define $\text{VC}^*(\mathbf{C}) = \text{VC}(\mathbf{C}^*)$.

Many classical results in machine learning theory center around computing bounds on the VC dimension of various classes of set systems. Below are some examples:

⁸The best known hardness result is actually somewhat stronger than **NP-hardness**. They show that approximating VC dimension within a factor of $2 - \epsilon$ is complete for Σ_3^P —a complexity class in the polynomial hierarchy that contains **NP**.

2.4. MEASURING AND EXPLOITING THE COMPLEXITY OF SET SYSTEMS

- Priority matrices, as defined in Definition 2.3.23, are examples of matrices that have VC dimension and VC codimension equal to 1.
- \mathbb{R}^n -**Halfspace-SC** and \mathbb{R}^n -**Ball-SC** both have VC dimension $n+1$ [Dud78].
- \mathbb{R} -**Interval-SC** has VC dimension 2, \mathbb{R}^2 -**Rectangle-SC** has VC dimension 4, and in general, \mathbb{R}^n -**2n-Sided-Box-SC** has VC dimension and VC codimension both equal to $2n$ [VC71].
- **Tree-SC** can be shown to have VC dimension and VC codimension equal to 2 using exactly the same method as for \mathbb{R} -**Interval-SC**.

As it turns out, all of the classes of set systems we have discussed in this thesis have *bounded VC dimension*. There is a very good reason for this:

Proposition 2.4.6. *Let \mathbf{C} be a hereditary class of set systems that has unbounded VC dimension. Then \mathbf{C} contains every set system matrix.*

Proof. If \mathbf{C} is hereditary and has unbounded VC dimension, then \mathbf{C} contains V_d and all of its submatrices, for all integers d . Let A be an arbitrary set system matrix of dimension $M \times N$, recalling that set system matrices contain no duplicate rows or columns. This immediately implies that A is a submatrix of V_M (modulo row and column reordering) obtained by deleting columns, from which we can conclude that $A \in \mathbf{C}$. The result follows since A was arbitrary. \square

It follows that hereditary problems with unbounded VC dimension inherit all of the hardness of the **Min-Set-Cover** problem and thus do not admit any improved approximation algorithms over general **Min-Set-Cover**.

VC dimension and codimension have many nice properties:

Proposition 2.4.7. *Let \mathbf{C} and \mathbf{D} be classes of set systems, and let A be a set system matrix. The following hold:*

1. *A (respectively \mathbf{C}) has finite VC dimension if and only if A (respectively \mathbf{C}) also has finite VC codimension. In particular, $\text{VC}^*(A) \leq 2^{\text{VC}(A)}$.*
2. *If $\mathbf{C} \subseteq \mathbf{D}$, then $\text{VC}(\mathbf{C}) \leq \text{VC}(\mathbf{D})$ and $\text{VC}^*(\mathbf{C}) \leq \text{VC}^*(\mathbf{D})$.*
3. *$\text{VC}(\mathbf{D}^P) = \text{VC}(\mathbf{D}) + 1$ and $\text{VC}^*(\mathbf{D}^P) = \text{VC}^*(\mathbf{D}) + 1$.*

Proof. 1. This property follows from the fact that V_d^T has 2^d distinct rows and thus is a submatrix of V_{2^d} . See also [VC71].

2. This is immediate from the definitions.

3. To show that $\text{VC}(\mathbf{D}^P) \geq \text{VC}(\mathbf{D}) + 1$, we simply note that we can write

$$V_{d+1} = \left[\begin{array}{c|c} V_d & V_d \\ \hline \mathbf{1}_{1 \times 2^d} & \mathbf{0}_{1 \times 2^d} \end{array} \right] = \left[\begin{array}{c|c} V_d & V_d \\ \hline \mathbf{1}_{1 \times 2^d} & \mathbf{1}_{1 \times 2^d} \end{array} \right] \wedge \left[\begin{array}{c|c} \mathbf{1}_{d \times 2^d} & \mathbf{1}_{d \times 2^d} \\ \hline \mathbf{1}_{1 \times 2^d} & \mathbf{0}_{1 \times 2^d} \end{array} \right],$$

2. FOUNDATIONS

which is the bitwise AND of a matrix of VC dimension d and a priority matrix. It follows that if some $A \in \mathbf{D}$ contains V_d as a submatrix, then some $A \wedge \Pi$ in \mathbf{D}^P contains V_{d+1} as a submatrix.

A proof of the fact that $\text{VC}(\mathbf{D}^P) \leq \text{VC}(\mathbf{D}) + 1$ can be found in [BKS11]; we outline the idea here. Let $d = \text{VC}(\mathbf{D}^P)$ and suppose, for a contradiction, that V_{d+2} is a submatrix of some matrix in \mathbf{D}^P . This assumption implies that $V_{d+2} = A \wedge \Pi$ for some priority matrix Π and some matrix A of VC dimension at most d . Let row r be the row of highest priority demand in A , and note that whenever $[A \wedge \Pi]_{r,j} = 1$, then column j of Π must consist entirely of ones. Hence if B is the submatrix of $A \wedge \Pi$ obtained by deleting all columns that do not contain a 1 in row r , then B is also a submatrix of A . We therefore obtain a contradiction, since B contains a copy of V_{d+1} , but we assumed $\text{VC}(A) \leq d$.

We have thus proven that $\text{VC}(\mathbf{D}^P) = \text{VC}(\mathbf{D}) + 1$. The dual version then follows from Proposition 2.3.29, since:

$$\text{VC}^*(\mathbf{D}^P) = \text{VC}((\mathbf{D}^P)^*) = \text{VC}((\mathbf{D}^*)^P) = \text{VC}(\mathbf{D}^*) + 1 = \text{VC}^*(\mathbf{D}) + 1$$

□

Part 3 of Proposition 2.4.7 has the following nice corollary:

Proposition 2.4.8. \mathbb{R}^n - $(n+k)$ -**Sided-Box-SC** has VC dimension $n+k$.

Proof. Immediate from the fact that \mathbb{R}^k - $2k$ -**Sided-Box-SC** has VC dimension $2k$, along with repeated applications of Proposition 2.3.28 and Proposition 2.4.7. □

We shall also provide a VC dimension result for the following covering problem:

COVERING PROBLEM: **Graph-Path-SC**
 ELEMENTS: The set E of edges of a graph $G = (V, E)$
 SETS: A family of paths in G

Proposition 2.4.9. **Graph-Path-SC** has unbounded VC dimension.

Proof. Consider the following counterexample. Construct a graph G_d by taking a path P of length d and replacing every edge by a triangle to obtain a chain of d triangles, each sharing a single vertex with the next. Let P' be the path of length d in G_d that passes through all vertices of P . It is clear that for every subset of the edges in P' , there exists a path in G_d containing those edges, and no other edges in P . This implies that these edges can be shattered, and thus $\text{VC}(G_d) = d$. Thus **Graph-Path-SC** contains instances of arbitrarily high VC dimension. □

Although it is not a tree, the counterexample we give in the proof of Proposition 2.4.9 is an extremely simple graph with many nice properties. It is a

so-called *series-parallel graph*—a natural generalization of trees obtained by iteratively replacing edges with paths or sets of parallel edges. It also has treewidth 2, pathwidth 2, bandwidth 2, and is outerplanar. Proposition 2.4.9 implies that we should not hope that our approximability results for trees can be extended to series-parallel graphs or other related simple graph classes. Indeed, covering series-parallel graphs with paths is no easier than general **Min-Set-Cover**.

Despite the fact that bounded VC dimension is a necessary condition for $o(\log M)$ approximation algorithms to exist, it alone is not known to be sufficient to obtain them. With our current knowledge, we need additional structural properties to hold in order to obtain better approximation guarantees.

2.4.2 ϵ -Nets and the Theorem of Haussler and Welzl

An intriguing connection between VC dimension and covering problems was discovered by Haussler and Welzl in 1986 [HW87]. They showed that all set systems of bounded VC dimension admit relatively small hitting sets for subproblems where every set contains a large number of elements. These hitting sets are known as ϵ -nets.⁹

Definition 2.4.10. Given a set system (X, \mathcal{S}) , a point $x \in X$ has *depth* L if it is contained in exactly L sets in \mathcal{S} . Equivalent, the *depth* of a row of a set system matrix A is the number of ones it contains. A matrix A is *L -deep* if all of its rows have depth L or greater.¹⁰

Definition 2.4.11. Given a set system (X, \mathcal{S}) , a subfamily of sets $\mathcal{C} \subseteq \mathcal{S}$ is an ϵ -net whenever every point $x \in X$ having depth ϵN or greater is contained in at least one set in \mathcal{C} , where $N = |\mathcal{S}|$. Equivalently, given a set system matrix A , an ϵ -net is a set cover for the submatrix of A obtained by deleting all rows containing fewer than ϵN ones.

An ϵ -net is simply a set cover that needs only to cover points that are contained in at least an ϵ -fraction of the available sets in \mathcal{S} . The remaining points can be discarded and ignored.

Historically, ϵ -nets were defined first for hitting set problems rather than set cover problems, and what we described in Definition 2.4.11 is theoretically an ϵ -net for the *dual set system* $(X, \mathcal{S})^*$. However, the literature tends not to distinguish between ϵ -nets for primal and dual set systems when discussing covering problems, instead relying on context to determine what is meant. Generally speaking, the convention used both in the literature and in this thesis is as follows:

⁹The geometric ϵ -nets we discuss here should not be confused with the ϵ -nets from probability theory or the theory of metric spaces, which share the same name but are entirely unrelated.

¹⁰The terminology employed in the literature for these notions is somewhat confusing. We use the term *depth* L only for points (meaning depth *exactly* L) and *L -deep* only for matrices (meaning all rows have depth *at least* L).

2. FOUNDATIONS

Remark 2.4.12. When discussing a covering problem in which the goal is to select a min-cardinality or min-cost set of *things* in order to (cover, hit, dominate, stab, etc.) a group of *objects*, an ϵ -net is a subset of *things* that (covers, hits, dominates, stabs, etc.) all of the *objects* that are incident to an ϵ -fraction of the *things*. In the matrix world, an ϵ -net is always a subset of the columns of a set system matrix, whose selection covers all rows of depth at least ϵN , where N is the number of columns.

For example, when discussing \mathbb{R}^2 -**Rectangle-SC**, an ϵ -net is a collection of rectangles that covers every point of depth ϵN or greater, where N is the number of rectangles. When discussing \mathbb{R}^2 -**Rectangle-HS**, an ϵ -net is a collection of points that hits every rectangle containing at least ϵN points, where N is the number of points.

The key contribution of Haussler and Welzl was to show that small ϵ -nets can be found for all unweighted covering problems having finite VC dimension. Translated into our language, their celebrated theorem is as follows:

Theorem 2.4.13 (Haussler and Welzl, 1986). *Let A be a set system matrix having VC codimension at most d . Then A admits an ϵ -net of size*

$$s = O\left(\frac{d}{\epsilon} \log \frac{d}{\epsilon}\right).$$

Moreover, a uniformly random sample of s columns from A is, with high probability, an ϵ -net.

Remark 2.4.14. We observe the following:

- Since all problems with finite VC dimension also have finite VC codimension (by Proposition 2.4.7), Theorem 2.4.13 effectively yields ϵ -nets of size $O\left(\frac{1}{\epsilon} \log \frac{1}{\epsilon}\right)$ for all set systems of bounded VC dimension.
- We note that the bound on the size of the ϵ -net is independent of the number of elements or sets in the set system, and depends only on the VC codimension d and the parameter ϵ itself.
- Since all submatrices of a matrix A also have VC codimension at most d , Theorem 2.4.13 also applies to all submatrices of A .
- Theorem 2.4.13 implies the existence of a randomized polynomial time algorithm to find a feasible covering of size s for instances in which every element lies within at least an ϵ sets. An ϵ -net can be found in expected polynomial time by simply randomly choosing a sample of columns of A repeatedly.

The result of Theorem 2.4.13 was improved many times in the years that followed. Blumer et al. reproved Theorem 2.4.13 with $s = O\left(\frac{d}{\epsilon} \log \frac{1}{\epsilon}\right)$ [BEHW89]. Soon thereafter, Komlós, Pach, and Woeginger provided a proof that eliminated the leading constant term, obtaining the result for $s = (1 + o(1))\left(\frac{d}{\epsilon} \log \frac{1}{\epsilon}\right)$ [KPW92]. In the same paper, they showed that this bound was essentially optimal by providing a randomized set system construction admitting no ϵ -nets of size less than $K\left(\frac{1}{\epsilon} \log \frac{1}{\epsilon}\right)$ for some positive constant K .

2.4.3 From ϵ -Nets to Set Covers

Since the paper of Haussler and Welzl, interest in ϵ -nets has grown significantly. Many authors have provided algorithms to construct ϵ -nets of smaller sizes than those guaranteed by the Haussler-Welzl method for specific set systems. Such algorithms are known as *ϵ -net-finders*:

Definition 2.4.15. Given a class \mathbf{C} of set systems, an *ϵ -net-finder of size s* for \mathbf{C} is a polynomial algorithm that, when given an instance $A \in \mathbf{C}$, produces an ϵ -net for A containing at most s sets.

As it turns out, ϵ -net-finders for many problems, including \mathbb{R}^2 -**Disk-SC**, \mathbb{R}^2 -**Disk-HS**, and \mathbb{R}^n -**Halfspace-SC**, have been constructed to yield ϵ -nets of size $s = O(\frac{1}{\epsilon})$, beating the $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ bound obtained directly from Theorem 2.4.13 [MSW90, Mat92, CV07] (numerous other ϵ -net finders, many of which are designed for specific covering problems, are overviewed in Chapter 3).

At a glance, ϵ -nets appear to have a key drawback—used naïvely, they only guarantee improved results for set systems in which every element is of sufficiently high depth. Fortunately, an important breakthrough occurred in 1994, when Brönnimann and Goodrich found a method of applying ϵ -net-finders to obtain good solutions to general set cover problems [BG95]. Their algorithm, known as the *iterative reweighting* method, achieves the following:

Theorem 2.4.16 (Brönnimann and Goodrich, 1994). *Let \mathbf{C} be a class of set systems admitting an ϵ -net-finder of size $s(\frac{1}{\epsilon})$ where s is an arbitrary function. Then there is an LP-relative polynomial algorithm for the unweighted **Min-Set-Cover** problem on \mathbf{C} that produces covers of size at most $4s(OPT)$, where OPT is the size of a minimum (fractional) cover.*

The original method used to prove this theorem calls the ϵ -net-finder multiple times on progressively refined versions of the input covering instance in which the multiplicity of some sets is increased. Along with Theorem 2.4.13 of Haussler and Welzl, Theorem 2.4.16 immediately implies the following:

Corollary 2.4.17. *All unweighted covering problems of bounded VC dimension admit a polynomial-time $O(\log OPT)$ -approximation.*

In some cases, this represents a significant improvement over the $O(\log M)$ -approximation obtained via the greedy algorithm.

Perhaps more importantly, the method of Brönnimann and Goodrich yields improved approximation algorithms for covering problems admitting improved ϵ -net-finders:

Corollary 2.4.18. *All unweighted covering problems admitting an ϵ -net-finder of size $s = O(\frac{1}{\epsilon} \phi(\frac{1}{\epsilon}))$ admit an $O(\phi(OPT))$ -factor approximation. In particular, an ϵ -net-finder of size $O(\frac{1}{\epsilon})$ yields a constant approximation.*

Theorem 2.4.16 was reproven a decade later by Even, Rawitz, and Shahar using a very elegant and simple linear-programming-based method [ERS05] (see

2. FOUNDATIONS

also [Lon01]). Their proof is very short, and they also manage to remove the leading factor of 4 in the approximation guarantee. We wholeheartedly recommend their paper as it presents, in an incredibly clear and concise manner, the connection between ϵ -net finders and LP-relative approximation algorithms for covering problems.

In 2010, Varadarajan extended the result of Even, Rawitz, and Shahar to weighted covering problems [Var10] in the event that the ϵ -net finder exhibits a property known as *quasi-uniformity*:

Definition 2.4.19. A randomized ϵ -net-finder of size s is *quasi-uniform* if it produces ϵ -nets in which each set is randomly chosen with probability at most $\frac{c}{N}s$ for some constant $c > 0$.

We note that in a uniformly chosen ϵ -net (such as those used to prove Theorem 2.4.16), each element is chosen with probability $\frac{1}{N}s$. Quasi-uniformity allows for a randomized ϵ -net finder in which each set's probability of being chosen may exceed this by some fixed multiplicative constant c .

Quasi-uniformity turns out to be useful when dealing with weighted covering problems. Using it, we can state Varadarajan's version of Theorem 2.4.16 for weighted covering problems. We include a brief proof for completeness, as it is not difficult (more details can be found in [Var10]):

Theorem 2.4.20. *Let $\phi(N)$ be a nondecreasing function (possibly a constant) and suppose that a hereditary class of set systems \mathbf{C} admits a quasi-uniform ϵ -net finder \mathcal{F} of size $\frac{1}{\epsilon}\phi(N)$, where $\phi(N) \in O(\log N)$.¹¹ Then there exists an expected polynomial-time $O(\phi(N))$ -approximation for the weighted covering problem on \mathbf{C} .*

Proof. Suppose that our goal is to approximate the minimum weight set cover problem on an $M \times N$ set system matrix A with weights \mathbf{w} . We reduce this to the problem of computing a small cover of a related $M/2$ -deep covering instance, and employ our ϵ -net finder.

We first solve the LP relaxation (SCLP) of the problem, and let \mathbf{x}^* be an optimal basic solution. Standard properties of basic solutions immediately imply that the support of \mathbf{x}^* (the set of positive entries) has size at most M . We create a set family \mathcal{S}^* , by including $\lfloor 2M \cdot x_S^* \rfloor$ copies of each set $S \in \mathcal{S}$ with $x_S^* \geq 1/(2M)$; *small* sets S with $x_S^* < 1/(2M)$ are not included. For each element $e \in X$ we now have

$$\sum_{S: x_S^* \geq \frac{1}{2M}, e \in S} \lfloor 2M \cdot x_S^* \rfloor \geq M \sum_{S: x_S^* \geq \frac{1}{2M}, e \in S} x_S^* \geq \frac{M}{2}, \quad (2.4.1)$$

where the second inequality uses the fact that small sets supply at most a $1/2$ unit of coverage for each element e . Let A^* be the set system matrix for set

¹¹We observe that in all cases, we must have $\frac{1}{\epsilon} \leq N$ (otherwise the problem of computing an ϵ -net is trivial, since no element can have depth greater than N). Thus by having ϕ as a function of N rather than $\frac{1}{\epsilon}$, we are in fact weakening the requirements of the theorem, allowing ϵ -net finders whose performance may vary according to an instance-specific function of N .

family \mathcal{S}^* and elements X , and assume that it has N^* columns. Equation (2.4.1) shows that A^* is $M/2$ -deep; we henceforth let $L := M/2$ so that A^* is L -deep. We let \mathbf{w}^* be a set of weights for A^* , inherited from A (so each copy of a column in A is given the same weight in A^*).

We now take $\epsilon = 1/L$ and employ our ϵ -net finder \mathcal{F} to A^* (note that \mathcal{F} can be applied to A^* because we assumed that \mathbf{C} was hereditary). The ϵ -net finder \mathcal{F} shall produce a covering of X in which each set in \mathcal{S}^* is included with probability $O\left(\frac{\phi(N^*)}{L}\right)$. The expected weight of the covering is then

$$\begin{aligned} O\left(\frac{\phi(N^*)}{L}\right) \cdot \mathbf{w}(\mathcal{S}^*) &\leq O(2\phi(N^*)/M) \cdot \sum_{S \in \mathcal{S}} [2M \cdot w_S x_S^*] \\ &\leq O(\phi(N^*)) \cdot OPT \leq O(\phi(N)) \cdot OPT, \end{aligned}$$

where the final inequality follows from the fact that $\phi(N) \in O(\log N)$. By running \mathcal{F} until an ϵ -net of desired weight is found, we can indeed obtain a covering of cost at most $\phi(N)OPT$ in expected polynomial time, completing the proof. \square

One should note that a few $O(1)$ factors are lost over the approximation obtained in [ERS05] for unweighted set cover. It remains unclear whether these losses can be avoided.

A noteworthy difference in Varadarajan's version for weighted set cover is that the approximation factor is $O(\phi(N))$, not $O(\phi(OPT))$ as in the original theorem of Brönnimann and Goodrich. Since the value of an optimal solution to a weighted covering problem may depend entirely on the weights used, we are unable to bound the approximation factor by anything involving the size of the optimal solution.

The Brönnimann–Goodrich theorem and related results essentially show that the existence of a good ϵ -net finder is sufficient to imply the existence of a good LP-relative algorithm for covering problems. There is a natural ‘reverse direction’ that follows quite easily: if good ϵ -nets do not exist, then the linear programming relaxation (SCLP) has a poor integrality gap, and thus good LP-relative algorithms cannot exist either.

Theorem 2.4.21. *Let \mathbf{C} be a hereditary class of set systems. Suppose for some $\epsilon > 0$ there is a matrix $A \in \mathbf{C}$ with no ϵ -net of size $\frac{k}{\epsilon}$. Then \mathbf{C} has integrality gap at least k .*

Proof. Suppose such an A exists and let B be the submatrix of A obtained by deleting all rows of depth less than $N\epsilon$, where N is the number of columns of A . Then B has no set cover containing fewer than $\frac{k}{\epsilon}$ sets (otherwise such a cover would be an ϵ -net for A). However, B is $N\epsilon$ -deep and thus taking each column of B with weight $\frac{1}{N\epsilon}$ produces a valid fractional covering for B . The cost of such a fractional covering is $N \cdot \frac{1}{N\epsilon} = \frac{1}{\epsilon}$, from which it follows that the integrality gap of the natural linear programming relaxation (SCLP) for B is at least k . But $B \in \mathbf{C}$ since \mathbf{C} is hereditary, and thus \mathbf{C} has integrality gap at least k . \square

2. FOUNDATIONS

In summary, good LP-relative approximation algorithms exist for covering problems if and only if good ϵ -net-finders do. See [ERS05] and [BKS11] for further discussion on the connection between integrality gaps and ϵ -nets.

Chapter 3

Related Work

In this chapter, we discuss a variety of prior results relating to covering problems having geometric or combinatorial structure. We stop short of attempting a full encyclopedic cataloguing of all known upper and lower bounds on approximability for all covering problems; instead, we discuss the important classical results in the area as well as all of the recent results related to our own work. In cases where our results improve upon the work of others, we briefly describe the significance of our contributions. Most of the results we discuss are for geometric covering problems; the idea of priority covering was developed more recently and is less widely known, so there are fewer results to report.

3.1 Known Hardness Results

Hardness results for covering problems go back many decades. The weakest type of hardness we shall discuss is **NP**-hardness. Of course, **Min-Set-Cover** and **Min-Hitting-Set** were among the first problems ever found to be **NP**-complete [Kar72]. Additionally, the **NP**-completeness of **Min-Vertex-Cover** showed that **Min-Set-Cover** remains hard even when each element lies in at most 2 sets. In the decades that followed, many other **NP**-hardness results followed for more restrictive covering problems via increasingly sophisticated chains of reductions. Eventually, **NP**-hardness was established for almost all nontrivial geometric covering and hitting set problems. We state the following:

Theorem 3.1.1. *The following covering problems are **NP**-hard, even in the unweighted case:*

- \mathbb{R}^2 -**Unit-Disk-SC**, \mathbb{R}^2 -**Unit-Disk-DS**, and \mathbb{R}^2 -**Unit-Square-SC** [HM85, CCJ90].
- \mathbb{R}^2 -**Unit-Square-HS** and \mathbb{R}^2 -**Unit-Disk-HS** by the self-duality of the above.
- \mathbb{R}^3 -**Halfspace-SC** and \mathbb{R}^3 -**Halfspace-HS** from the previous results via the standard lifting transformation (Proposition 2.3.17).

3. RELATED WORK

- **Rectilinear-Polygon-Cover**, even when the rectilinear polygon contains no holes [CR94].

Many additional **NP**-hardness results were published for other geometric covering problems, but with the subsequent introduction of the stronger notions of **APX**-hardness and **MAX-SNP**-hardness (both of which rule out the possibility of a PTAS unless $\mathbf{P} = \mathbf{NP}$ [PY91]), most of these **NP**-hardness results have been superseded. The problems mentioned in Theorem 3.1.1 are essentially the only remaining problems for which we know of no stronger hardness results. As we shall see, all except **Rectilinear-Polygon-Cover** are known to admit a PTAS in the unweighted case.

As for **APX**-hardness and **MAX-SNP**-hardness, a variety of results are known. One of the first key breakthroughs was for the **Rectilinear-Polygon-Cover** problem, which was proven to be **MAX-SNP**-hard by Berman and DasGupta using a complicated but incredibly beautiful sequence of gadgets that yield an L-reduction from a degree-bounded version of **Min-Vertex-Cover** that was known to be **MAX-SNP**-hard at the time. Their method yielded the first proof that **Rectilinear-Polygon-Cover** (and hence \mathbb{R}^2 -**Rectangle-SC**) does not admit a PTAS. Unfortunately, their construction requires the rectilinear polygon to contain holes; we know of no **APX**-hardness or **MAX-SNP**-hardness result for **Rectilinear-Polygon-Cover** in the event that the polygon is simple.

With the result of Alimonti and Kann showing that **3-Regular-Graph-SC** is **APX**-hard [AK00], it became much easier to establish **APX**-hardness for covering problems. Many hardness results for geometric problems such as \mathbb{R}^2 -**Triangle-Cover** follow immediately (as mentioned in Section 2.3.4). Har-Peled provides an incredibly simple and clever trick to extend this result to the case that all triangles are fat [HP09]. By using Vizing’s Theorem to 4-edge colour all 3-regular graphs, Har-Peled is able to encode **3-Regular-Graph-SC** using points and triangles in the plane that are all approximately right-angled. It follows that \mathbb{R}^2 -**Fat-Triangle-Cover** is **APX**-hard. In the same paper, Har-Peled also uses **3-Regular-Graph-SC** encoding to establish the **APX**-hardness of \mathbb{R}^2 -**Circle-SC**—the problem of covering points in the plane using circles (*boundaries* of disks). Har-Peled also applies the standard lifting transformation to extend the result to \mathbb{R}^3 -**Plane-SC**—covering points in three-dimensional space with planes (not *halfspaces*, but their boundaries).

In his PhD thesis, van Leeuwen gives several L-reductions from **3-Regular-Graph-SC** to other geometric covering problems, including \mathbb{R}^2 -**Rectangle-SC** and \mathbb{R}^2 -**Ellipse-Cover**, establishing their **APX**-hardness. However, his results do not appear to be applicable to fat rectangles or fat ellipses. We provide many stronger hardness results for geometric covering in Chapter 5.

As for non-geometric problems, a result of Cheriyan, Jordan, and Ravi can be used to show that **Tree-SC** is **APX**-hard, even if all of the paths form a cycle on the leaves of the tree [CJR99]. Though **Vertical-Tree-SC** is solvable in polynomial time, we do obtain an **APX**-hardness proof for **Priority-Vertical-Tree-SC** in Chapter 5.

3.2 ϵ -Net Finders for Geometric Set Cover

We recall Theorem 2.4.16, originally due to Brönnimann and Goodrich, which implies that ϵ -net-finders of size $s(\frac{1}{\epsilon})$ can be used to obtain set covers of size $O(s(OPT))$, yielding constant (or near-constant) approximations for covering problems admitting ϵ -nets of size linear (or nearly linear) in $\frac{1}{\epsilon}$. Since its introduction, a number of different authors have employed it to obtain improved approximation algorithms for covering problems by constructing ϵ -net-finders. In fact, there are so many results of this nature that it is infeasible to list them all here; we only mention a few of the more important contributions. Until recently, almost all of these results have applied only to the unweighted setting.

We begin by noting that Theorem 2.4.13—the key result of Haussler and Welzl—yields $O(\log OPT)$ -approximations for all covering problems of bounded VC dimension. Recall that when discussing VC dimension, we defined the matrix V_d and characterized a hereditary class of set systems \mathbf{C} according to the largest value of d for which $V_d \in \mathbf{C}$. One might wonder if any useful results can be obtained if V_d is replaced by a different canonical family of matrices. One such attempt, due to Ding, Seymour, and Winkler [DSW94], is to consider the $d \times \binom{d}{2}$ matrix K_d whose columns are all distinct and each contain exactly two ones (K_d is precisely the node-arc incidence matrix of the complete graph on d vertices, and is a submatrix of V_d). They show that if a matrix A contains no K_d submatrix for a fixed constant d , then there is a set cover for A of size polynomial in $\nu(A)$ —the maximum number size of an ‘independent set’ of rows of A (that is, the maximum size of a set of rows of A , no two of which contain a one in the same column).

A simple application of this theorem produces a result analogous to the Haussler-Welzl theorem (Theorem 2.4.13). If A is (ϵN) -deep (where N is the number of columns), then clearly $\nu(A) \leq \frac{1}{\epsilon}$. If A also contains no K_d submatrix for a fixed constant d , then the result of Ding, Seymour, and Winkler implies that A admits a set cover of size polynomial in $\frac{1}{\epsilon}$. Consequently, matrices that forbid sufficiently large K_d submatrices admit polynomial-sized ϵ -nets. Unfortunately, despite the interesting nature of this result and its connections to the combinatorics of hypergraphs, the ϵ -nets produced are simply too large to facilitate any improvements in approximability for geometric covering problems.

For covering and hitting set problems with disks in the plane and half-spaces in \mathbb{R}^3 , $O(\frac{1}{\epsilon})$ -size ϵ -net-finders were known many years ago [MSW90], long before the result of Brönnimann and Goodrich [BG95]. Consequently, the Brönnimann-Goodrich theorem immediately yielded constant approximations for \mathbb{R}^2 -**Disk-SC**, \mathbb{R}^2 -**Disk-HS**, \mathbb{R}^3 -**Halfspace-SC** and \mathbb{R}^3 -**Halfspace-HS**.¹ A key breakthrough came in 2005 via a result of Clarkson and Varadarajan that generalized these ϵ -net-finders to numerous other set systems. Clarkson and Varadarajan showed that small ϵ -nets (and therefore, small set covers), could be found for geometric set systems exhibiting low *union complexity* [CV07, Var09]. In an approximate sense, the union complexity of a collection of geometric ob-

¹More recently, these problems have been shown to admit a PTAS; see Section 3.4.

3. RELATED WORK

jects is proportional to the worst case combinatorial complexity of the boundary of the union of any subset of those objects. As illustrating examples, consider the following:

- The union of N half-planes contains only $O(N)$ segments along its boundary.
- The union of N rectangles in the plane can contain $O(N^2)$ segments along its boundary if, for example, the rectangles are laid out in a ‘plaid’ fashion.
- A non-trivial result is that an arrangement of N disks in the plane has union complexity $O(N)$ [KLPS86].

We shall not give the full, formal definition of union complexity here. It suffices to know that the aforementioned notion of combinatorial complexity is, to within a constant factor, equal to the union complexity of the set system [Var09], and can be generalized to higher dimensional Euclidean spaces. The pure, rigorous, nongeometric definition of complexity used by Clarkson and Varadarajan is somewhat cumbersome and more difficult to apply on its own; details can be found in the journal version of their paper [CV07].

The main result of Clarkson and Varadarajan essentially states that set systems having linear or near-linear union complexity admit polynomial-time approximation algorithms whose factor of approximation is constant or near-constant, respectively. A simplified statement of their contribution is as follows:

Theorem 3.2.1 (Clarkson and Varadarajan, 2005). *Let \mathbf{C} be a geometric covering set system and suppose there is a nondecreasing function f such that for all set systems $(X, \mathcal{S}) \in \mathbf{C}$, for all $k \geq 1$, and all families $C \subseteq X$ of objects with $|C| = k$, the union complexity of C is at most $kf(k)$. Then \mathbf{C} admits an ϵ -net finder of size $O(\frac{1}{\epsilon}f(\frac{1}{\epsilon}))$, and consequently, the unweighted covering problem on \mathbf{C} admits an LP-relative polynomial-time $O(f(OPT))$ -approximation.*

To prove Theorem 3.2.1, Clarkson and Varadarajan give an algorithm that produces ϵ -nets by first choosing a uniformly random group of objects to form a partial cover, and then, guided by structural properties related to the union complexity of the objects, performs a series of repair steps to cover the remaining uncovered deep points. By applying a counting lemma from [CS89], they are able to prove that the total number of sets used is, on average, not too large.

More recently, the approximability bound in Theorem 3.2.1 has been improved from $O(\frac{1}{\epsilon}f(\frac{1}{\epsilon}))$ to $O(\frac{1}{\epsilon}\log(f(\frac{1}{\epsilon})))$ using similar techniques [AES10, Var09]. This shaves off an additional logarithmic factor in cases where the union complexity is small but non-constant.²

Applications of Theorem 3.2.1 and its improvements are immediately apparent. For example, knowing that an arrangement of k disks in the plane has union complexity $O(k)$, we can immediately prove the existence of an

²Technically, for the result in [Var09] to hold, we need $f(k) \in \omega(\log^{(j)} k)$ for some constant j , where $\log^{(j)}$ is the logarithm iterated j times. However, the algorithm we give in Chapter 6 achieves a stronger result without this requirement.

$O(1)$ -approximation for unweighted \mathbb{R}^2 -**Disk-SC** via the Clarkson-Varadarajan method. Several new results have also been proven using similar techniques. In fact, the method is quite widely applicable in geometric set cover, in part because many results about the complexity of configurations of intersecting geometric objects are widely known. Below are a few examples:

- Boissonat et al. have proven that unit cubes in \mathbb{R}^3 have linear union complexity [BSTY98], implying that unweighted \mathbb{R}^3 -**Unit-Cube-SC** and \mathbb{R}^3 -**Unit-Cube-HS** both admit a constant approximation (the latter follows from the fact that \mathbb{R}^3 -**Unit-Cube-SC** is self-dual). We rule out a PTAS for these problems in Chapter 5.
- Unweighted \mathbb{R}^3 -**3-Sided-Box-SC** and its hitting set version \mathbb{R}^3 -**3-Sided-Box-HS** are subproblems of \mathbb{R}^3 -**Unit-Cube-SC** and \mathbb{R}^3 -**Unit-Cube-HS** and thus also admit a constant approximation.
- A paper of Efrat et al. [ERS93] proves linear union complexity for *fat wedges*—unbounded wedge-shaped regions in the plane whose boundary consists of two rays. Specifically, Efrat et al. show that if each wedge has an opening angle of at least δ , then the union complexity of k wedges is $O(k)$, with the constant in the big O varying with respect to δ . This implies that unweighted \mathbb{R}^2 -**Fat-Wedge-Cover**—the problem of covering points in the plane with fat wedges—admits a constant approximation. We rule out a PTAS for this problem in Chapter 5.
- Matoušek et al. have shown that k fat triangles of roughly the same size have $O(k)$ union complexity [MPS⁺94]; it follows that unweighted \mathbb{R}^2 -**Fat-Triangle-Cover** admits a constant approximation when all the triangles are roughly the same size. Of course, the **APX**-hardness result of Har-Peled implies that no PTAS is possible, even in this restricted case.
- More recently, Ezra et al. have proven that general fat triangles have union complexity $O(k2^{\alpha(k)} \log^*(k))$ where $\alpha(k)$ is the extremely slow-growing *inverse-Ackermann function* [EAS11]. Plugging this into the result of Clarkson and Varadarajan yields an $O(2^{\alpha(OPT)} \log^*(OPT))$ -approximation for general unweighted \mathbb{R}^2 -**Fat-Triangle-Cover**.³ We improve this to $O(\log \log^*(OPT))$ in Chapter 6.
- Using techniques involving Davenport-Schinzel sequences (see the text of Sharir and Agarwal [SA95] for a complete overview of the related theory), it can be shown that the downward shadow of k line segments in the plane has complexity $O(k\alpha(k))$, where $\alpha(k)$ is again the inverse-Ackermann function. This can be used to yield an $O(\alpha(OPT))$ -approximation for unweighted \mathbb{R}^2 -**Segment-Shadow-SC** (which we improve to $O(\log \alpha(OPT))$)

³More recently, with de Berg, these authors have apparently improved the bound to $O(k \log^* k)$, yielding an $O(\log^*(OPT))$ -approximation for unweighted \mathbb{R}^2 -**Fat-Triangle-Cover**.

3. RELATED WORK

in Chapter 5). Similar near-constant results can be obtained for downward shadows of k -intersecting functions for arbitrary k .

Using methods in Chapter 6, we can extend all of these results to weighted covering problems, with OPT replaced by N in the non-constant approximation guarantees.

Unfortunately, the Clarkson–Varadarajan method on its own has yielded no improvement for problems such as \mathbb{R}^2 -**Rectangle-SC** that do not exhibit nice union complexity. Moreover, these methods have not been very fruitful outside of the area of geometric covering problems. To formally obtain the bounds required for Clarkson and Varadarajan’s method to go through on, e.g., a non-geometric set system, one must construct something called a *configuration system* and prove non-trivial results about its structure. It is conceivable that, with significant effort, such a method could be used to obtain improvements for non-geometric covering problems. However, we do not know of any such result.

Aronov, Ezra, and Sharir recently provided some additional techniques to construct ϵ -nets for hitting set problems involving rectangular boxes in two and three dimensions [AES10]. Their technique is related to that of Clarkson and Varadarajan, but differs from it in several subtle ways. Their improvements also yield better results for some types of covering problems involving fat objects. We list a few results that follow from their work:

- Unweighted \mathbb{R}^2 -**Rectangle-HS** and its three-dimensional analogue \mathbb{R}^3 -**Sided-Box-HS** both admit an $O(\log \log OPT)$ -approximation.
- Hitting set involving arbitrary fat objects in the plane admits an approximation of factor $O(\log \log \log OPT)$.
- Covering involving general *locally fat objects* (and in particular, fat convex objects) in the plane admits an $O(\log \log OPT)$ -approximation.
- In the above example, if the locally fat objects are of similar sizes, then the approximation factor can be improved to a near-constant function of OPT using results of de Berg [dB10] and properties of Davenport-Schinzel sequences (see [SA95]).

In another recent paper, Pyrga and Ray have provided a way to reprove several of the ϵ -net results of Clarkson and Varadarajan without using union complexity at all, instead relying on elementary combinatorial and geometric arguments [PR08]. They provide a simple proof that \mathbb{R}^3 -**Halfspace-SC** admits ϵ -nets of size $O(\frac{1}{\epsilon})$. A few generalizations are also given.

Recently, Varadarajan [Var10] gave an elegant algorithm that extends some of the unweighted results above to the weighted setting. The algorithm produces randomized quasi-uniform ϵ -nets of small expected size using a novel sampling-based approach. Using this and a version of Theorem 2.4.20, Varadarajan is able to obtain an $O(2^{O(\log^* N)} f(N))$ -approximation for weighted covering problems having union complexity $O(kf(k))$ (where the leading exponential factor can be dropped if $f(k) \in \omega(\log^{(j)} k)$ for some constant j). We discuss Varadarajan’s

method in more depth in Chapter 6, where we give our own quasi-uniform sampling algorithm that makes various improvements, including removing the leading exponential term entirely from the approximation factor, simplifying the algorithm and its analysis, and providing applications to non-geometric set systems.

An important property of ϵ -net based methods is that they always yield LP-relative approximability results; that is, the approximability obtained is always relative to the standard LP relaxation (SCLP) (recall Definition 2.1.8). It follows that ϵ -net based methods cannot yield approximation algorithms whose factor of approximation is lower than the integrality gap of (SCLP), which is typically bounded away from 1 for all set systems except those that are totally unimodular. Consequently, ϵ -net based are unable to produce anything better than constant-factor approximations. Additionally, the constants involved are quite typically large. Alternate techniques are required to obtain PTAS results or exact algorithms. Moreover, as we shall see in the next section, ϵ -nets of size $o(\frac{1}{\epsilon} \log(\frac{1}{\epsilon}))$ need not always exist, even for relatively simple geometric set systems. Indeed, for some set systems, it is impossible to beat the naïve greedy-based logarithmic approximation for set cover using *any* LP-relative algorithm.

3.3 The Pach-Tardos Counterexample

For many decades, a prevailing conjecture amongst many computational geometers was that all sufficiently simple geometric covering problems admitted ϵ -nets of size linear in $\frac{1}{\epsilon}$ [AES10, MSW90]. This was settled in the negative by Alon, who gave a method of constructing set systems consisting of points and (*non*-axis aligned) slabs in the plane whose minimum ϵ -nets have size $\Omega(\frac{1}{\epsilon} \alpha(\frac{1}{\epsilon}))$, where α is the inverse-Ackermann function [Alo10]. However, it still remained open whether the $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ upper bound could be matched in a geometric set system of low VC dimension.

A recent construction due to Pach and Tardos has answered this in the affirmative by proving that \mathbb{R}^2 -**Rectangle-SC** admits no ϵ -nets of size $o(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ [PT11]. Instead, they show that the natural LP-relaxation (SCLP) for \mathbb{R}^2 -**Rectangle-SC** instances may exhibit a logarithmic integrality gap. Their construction, inspired by a recent hypergraph colouring result in combinatorics, is too complicated to describe here, but we shall outline their result and its consequences.

Theorem 3.3.1 (Pach and Tardos, 2010). *For all $\epsilon > 0$ and for all sufficiently large N , there exists a \mathbb{R}^2 -**Rectangle-SC** set system (X, \mathcal{S}) on N rectangles exhibiting all of the following properties:*

- \mathcal{S} consists entirely of axis-aligned rectangles in the plane whose boundaries intersect pairwise either zero times, or four times.
- (X, \mathcal{S}) has VC dimension 2 and VC codimension 2.

3. RELATED WORK

- The set X only contains points of depth at least ϵN (that is, each point in X is contained in at least ϵN sets in \mathcal{S}).
- (X, \mathcal{S}) admits no set cover of size smaller than $\frac{1}{9\epsilon} \log \frac{1}{\epsilon}$.

The proof of Theorem 3.3.1 explicitly constructs \mathbb{R}^2 -**Rectangle-SC** instances that admit no ϵ -nets of size $o(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$. Moreover, this difficulty persists even for instances with much lower VC dimension and codimension than typical \mathbb{R}^2 -**Rectangle-SC** instances. Since the boundaries of the rectangles in their construction always intersect pairwise either zero times or four times, the intersection graph of the rectangles is a comparability graph (and hence a perfect graph); even under this strong condition, small ϵ -nets can be ruled out.

An important aspect of the Pach-Tardos construction is that N can be as small as $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$, which is polynomial in $\frac{1}{\epsilon}$. Specifically, this implies that there are \mathbb{R}^2 -**Rectangle-SC** instances admitting no ϵ -net of size $o(\frac{1}{\epsilon} \log N)$. Via a direct application of Theorem 2.4.21, it follows that the integrality gap of \mathbb{R}^2 -**Rectangle-SC** instances may be as large as $\Omega(\log N)$ on instances having N rectangles. In particular, no LP-relative approximation algorithm for unweighted \mathbb{R}^2 -**Rectangle-SC** can beat the trivial greedy method by more than a constant factor (recalling that the greedy algorithm for set cover achieves an $O(\log N)$ -approximation). Consequently, the worst case integrality gap of \mathbb{R}^2 -**Rectangle-SC** is precisely $\Theta(\log N)$.

Via simple geometric encodings and self-duality, the Pach-Tardos counterexample can be generalized to other set systems:

Corollary 3.3.2. *The following set systems admit no ϵ -net finders of size $o(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ and hence have an integrality gap of $\Theta(\log N)$:*

- \mathbb{R}^3 -**4-Sided-Box-SC** and \mathbb{R}^4 -**4-Sided-Box-SC**.
- The dual problem \mathbb{R}^4 -**4-Sided-Box-HS**.
- \mathbb{R}^4 -**Halfspace-SC** and its dual problem \mathbb{R}^4 -**Halfspace-HS**.

Pach and Tardos also give a weaker ϵ -net lower bound for the hitting set version \mathbb{R}^2 -**Rectangle-HS** using a completely unrelated method:

Theorem 3.3.3 (Pach and Tardos, 2010). *There is a constant $C > 0$ such that for all $\epsilon > 0$ and for all sufficiently large N , there exists a \mathbb{R}^2 -**Rectangle-HS** set system (X, \mathcal{S}) on N rectangles exhibiting all of the following properties:*

- (X, \mathcal{S}) has VC dimension 3.⁴
- The set X only contains points of depth at least ϵN (that is, each point in X is contained in at least ϵN sets in \mathcal{S}).
- (X, \mathcal{S}) admits no set cover of size smaller than $C \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}$.

⁴In their paper, Pach and Tardos claim to have a different construction yielding the same result for a set system of VC dimension 2. However, they have deferred the proof to a future publication.

In a manner similar to what we discussed previously for Theorem 3.3.1, this establishes a lower bound of $\Omega(\log \log N)$ on the integrality gap of unweighted \mathbb{R}^2 -**Rectangle-HS**. Noteworthy is the fact that this asymptotically matches the approximation factor of the ϵ -net based algorithm of Aronov, Ezra, and Sharir for \mathbb{R}^2 -**Rectangle-HS**, firmly establishing the integrality gap of \mathbb{R}^2 -**Rectangle-HS** at $\Theta(\log \log N)$.

The counterexamples of Pach and Tardos demonstrate that VC dimension is not an ideal measure of set system complexity, and show that ϵ -net based methods have limitations. In particular, to make any additional progress on \mathbb{R}^2 -**Rectangle-SC** or \mathbb{R}^2 -**Rectangle-HS**, it will be necessary to employ some type of LP strengthening, or perhaps abandon LP-based methods altogether and pursue a different algorithmic idea. Nevertheless, no existing hardness result has ruled out a constant approximation for \mathbb{R}^2 -**Rectangle-SC** or \mathbb{R}^2 -**Rectangle-HS** using other techniques; determining whether or not a constant approximation is possible remains one of the largest open problems in the area.

3.4 The Local Search Approach

In light of the drawbacks of ϵ -net based methods, Mustafa and Ray recently proposed a different approach. They give a PTAS for a wide class of unweighted geometric hitting set problems via a *local search* technique [MR10]. Cast in our framework, their algorithm works roughly as follows: fix a constant k and take any feasible set cover \mathcal{C} . Whenever possible, find a family of k sets in \mathcal{C} that can be replaced by some family of $k - 1$ sets while still covering every object in the universe, making such replacements until no more are possible. For a fixed k , this runs in polynomial time. As Mustafa and Ray show, this method produces a $1 + O\left(\frac{1}{\sqrt{k}}\right)$ -approximation for hitting set problems on set systems satisfying certain locality conditions. By taking k to be sufficiently large, a $(1 + \epsilon)$ -approximation can be obtained for a wide variety of problems. Mustafa and Ray obtain a variety of results using this technique:

Theorem 3.4.1 (Mustafa and Ray, 2010). *The following covering problems admit a PTAS:*

- *Unweighted \mathbb{R}^3 -**Halfspace-HS**.*
- *Unweighted \mathbb{R}^2 -**Pseudodisk-HS**.*
- *As a special case of the above, unweighted \mathbb{R}^2 -**Square-HS**, \mathbb{R}^2 -**Disk-HS**, and any unweighted hitting set problem involving translated copies of identical convex regions in the plane.*

We observe that via simple encodings, the result of Mustafa and Ray can be used to obtain a PTAS for a few other covering problems:

Corollary 3.4.2. *The following covering problems admit a PTAS:*

3. RELATED WORK

- *Unweighted \mathbb{R}^3 -Lower-Halfspace-SC* (from the PTAS result for \mathbb{R}^3 -Halfspace-HS via the geometric duality result described in Proposition 2.3.18).
- *Unweighted \mathbb{R}^2 -Disk-SC* (from the previous point, via the lifting transformation described in Proposition 2.3.17).
- *Any unweighted set cover problem involving translated copies of identical convex regions in the plane* (by self-duality and the third point in Theorem 3.4.1).

The results of Mustafa and Ray currently do not seem applicable to set cover with general pseudodisks in the plane. In particular, it is unclear whether or not their result can be extended to the geometric covering problem involving squares in the plane having different sizes, as no duality result relates this problem to \mathbb{R}^2 -Square-HS. The existence of a PTAS for \mathbb{R}^2 -Pseudodisk-SC and \mathbb{R}^2 -Square-SC remains an open question. Additionally, it is unclear whether any results for weighted covering problems can be obtained via the local search approach.

In Chapter 5, we provide several **APX**-hardness proofs that demonstrate that the PTAS methods of Mustafa and Ray are unlikely to admit generalization to more difficult set systems. In particular, no PTAS exists for covering or hitting set problems involving half-spaces in four dimensions, cubes and unit spheres in three dimensions, or regions in the plane whose boundaries intersect pairwise at most three times.

3.5 Other Methods for Geometric Set Cover

Here, we shall briefly describe a few other relevant techniques that have been used to obtain solutions or approximations for geometric covering problems. Most of these methods rely heavily on restrictive geometric properties of specific set systems, and consequently, they are difficult to generalize.

The *piercing problem* is a special case of unweighted geometric hitting set in which a family of geometric objects \mathcal{S} is provided but no set of points X is given; instead, *any* point may be used to hit the objects. Like **Rectilinear-Polygon-Cover**, piercing problems are *not* hereditary (because columns of a set system matrix cannot be removed). Piercing is **NP**-hard with unit squares and unit disks in the plane [FPT81], but admits a PTAS in a much more general set of cases than ordinary hitting set. We list a few of the key results in the area:

- Hochbaum and Maass show that in any fixed dimension, there is a PTAS for piercing problems involving fat objects of approximately equal size [HM85].
- By duality, the previous result implies the existence of a PTAS for the problem of covering a fixed set of points in any Euclidean space using a minimum number of identical translated copies of a single fat object, where the objects may be freely placed at any location but not rotated.

- Using planar separator theorems and their higher-dimension generalizations, Chan gives a PTAS for piercing problems involving fat objects of arbitrary size in any fixed dimension [Cha03].

We know of only one true PTAS result for the *weighted* version of an **NP**-hard geometric covering problem. The weighted version of **\mathbb{R}^2 -Unit-Square-SC** (and by self-duality, **\mathbb{R}^2 -Unit-Square-HS**) admits a PTAS due to Erlebach and van Leeuwen, who employ an approach involving shifted grids [EvL10] (see also the PhD thesis of van Leeuwen [vL09]). A similar method can also be used to obtain a PTAS for the weighted version of the dominating set problem **\mathbb{R}^2 -Unit-Square-DS**.

A weaker PTAS-like result is that of Har-Peled and Lee, who give a PTAS for weighted covering in the plane with *any* class of fat objects, provided that each object is allowed to expand by a tiny fraction of its diameter [HPL08]. In Chapter 5, we give **APX**-hardness results for several covering problems involving fat objects, showing that without allowing this expansion, a PTAS cannot be obtained.

We know of few non-trivial examples of geometric covering problems that are known to be polynomial-time solvable. Har-Peled and Lee give an exact dynamic programming algorithm for weighted **\mathbb{R}^2 -Halfplane-SC** (and, by implication, **\mathbb{R}^2 -Halfplane-HS**) [HPL08]. Their method runs in $O(n^5)$ time on an instance containing n points and half-planes. In Chapter 4, we give a more general dynamic programming algorithm that achieves the same result, but reduces the running time by a factor of n .

Ambühl et al. give a polynomial time dynamic programming algorithm for weighted covering of points in a narrow strip using unit disks [AEMN06]; their method appears to be unrelated to ours or that of Har-Peled and Lee.

Chapter 4

Polynomial-Time Algorithms for Geometric Set Cover

In this chapter, we examine some geometric covering problems in the plane that exhibit a very special kind of structure that allows for exact solvability in polynomial time via dynamic programming algorithms. All of the algorithms in this section work for both weighted and unweighted covering problems. Additionally, they can be configured to either output the minimum cost of a cover, or to output a minimum-cost cover itself.

Most of these results were first presented in [CG11], building upon an earlier algorithm for \mathbb{R}^2 -**3-Sided-Box-SC** that appeared in [CGK10a]. This algorithm relied on a somewhat messy recursion involving shortest path subproblems, and has been rendered obsolete by our newer proof, which we believe is much cleaner, simpler, and more general.

4.1 Main Dynamic Programming Algorithm

Our main tool in this chapter is a dynamic programming algorithm for a specific covering problem that can fully encode many other problems. We begin by describing this problem, which we shall call \mathbb{R}^2 -**2-Intersecting-Shadow-SC**.

For a set Y of points in the plane, we recall that the *downward shadow* of Y is the set of all points (a, b) such that there is a point $(a, y) \in Y$ with $y \geq b$. We shall obtain an exact algorithm for a geometric covering problem involving downward shadows of infinite curves in the plane. Whenever we say *curves* in this context, we specifically mean simple *Jordan arcs*—continuous curves that are *not* self-intersecting. See [dBCvKO08] for further details.

Definition 4.1.1. A family \mathcal{F} of curves is *k-intersecting* if, for all $C_1, C_2 \in \mathcal{F}$ with $C_1 \neq C_2$, the number of points common to C_1 and C_2 is at most k .

Definition 4.1.2. A curve C is x -monotone if, for every $x \in \mathbb{R}$, the set of all y such that $(x, y) \in C$ is a convex subset of \mathbb{R} (i.e. an interval). In less formal terms, one may walk along the entirety of C without ever moving to the left.

We finally give the formal definition of our key problem:

COVERING PROBLEM: \mathbb{R}^2 -**2-Intersecting-Shadow-SC**
 ELEMENTS: A finite subset of \mathbb{R}^2
 SETS: A family of downward shadows of pairwise 2-intersecting infinite x -monotone curves in \mathbb{R}^2 .

We add the ‘infinite’ here because of the possibility that two curves may intersect at most twice, but have downward shadows whose *boundaries* intersect more than twice. We require the boundaries of the sets in \mathcal{S} to be 2-intersecting, not just the curves. For infinite x -monotone curves, the boundary of the downward shadow of the curve is simply the curve itself.

We state our main result:

Theorem 4.1.3. *There exists a polynomial-time exact algorithm for weighted \mathbb{R}^2 -2-Intersecting-Shadow-SC. Moreover, it requires $O(NM^2(M+N))$ primitive operations to run on a set system consisting of M points and N regions*

For the purposes of Theorem 4.1.3, it suffices to allow a ‘primitive operation’ to be one of the following:

- Adding or comparing two coordinates or weights,
- Determining if a given point lies above or below a given curve,
- Determining the coordinates of the points of intersection of a pair of curves.

For the remainder of this section, we shall assume that such operations can each be completed in $O(1)$ time. Note that even if arbitrary arithmetic operations are allowed in $O(1)$ time, more time may be required for the primitive operations if, for example, the x -monotone curves are piecewise-linear functions each consisting of k pieces.

Before proceeding to the proof, we shall develop some simple theory, from which a proof shall immediately present itself. We wish to give a polynomial-time dynamic programming algorithm for the weighted cover of a finite set of points $X \subseteq \mathbb{R}^2$ by a set \mathcal{S} of downward shadows of 2-intersecting x -monotone curves C_1, \dots, C_N . For $1 \leq i \leq N$, define the region $S_i \in \mathcal{S}$ to be the downward shadow of the curve C_i and let it have positive cost w_i . As usual, we shall let $M = |X|$.

For simplicity in the presentation of our proof, we shall make the following assumptions:

- Each curve C_i is the graph of a smooth univariate function with domain $[-\infty, \infty]$.

4.1. MAIN DYNAMIC PROGRAMMING ALGORITHM

- All intersections are transverse (that is, no pair of curves intersect tangentially).
- No points in X lie on any curve C_i .

It is not difficult to get around these assumptions by, if necessary, perturbing each curve slightly while maintaining the element-set incidence relation. However, we retain them to keep our exposition as clean as possible.

We shall abuse notation by writing $C_i(x)$ for the unique $y \in \mathbb{R}$ such that (x, y) lies on the curve C_i . We say curve C_i is *wider* than curve C_j (written $C_i \succ C_j$) whenever $C_i(x) > C_j(x)$ for all sufficiently large x . We may also write $S_i \succ S_j$ whenever $C_i \succ C_j$. We note that \succ is a total ordering and thus we can order all curves by width, so we assume without loss of generality that $C_i \succ C_j$ whenever $i > j$. The width-based ordering of curves is useful because of the following key observation:

Proposition 4.1.4. *If $C_i \succ C_j$, then $S_j \setminus S_i$ is connected (topologically).*

Proof. This is clearly true if C_i and C_j intersect once or less. If C_i and C_j intersect transversely twice—say, at (x_1, y_1) and (x_2, y_2) with $x_2 > x_1$ —then the area above C_i but below C_j can only be disconnected if $C_j(x) > C_i(x)$ for $x < x_1$ and $x > x_2$, implying $C_j \succ C_i$. \square

For all $1 \leq i \leq N$ and all intervals $[a, b]$, define $X[a, b]$ to be all points in X with x -coordinate in $[a, b]$, and define $X[a, b, i]$ to be $X[a, b] \setminus S_i$. Define $\mathcal{S}_{<i}$ to be the set $\{S_1, \dots, S_{i-1}\}$ of all regions of width less than S_i . Let $W[a, b, i]$ denote the minimum cost of a solution to the weighted set cover problem on the set system $(X[a, b, i], \mathcal{S}_{<i})$ (with weights inherited from the original problem). In other words, $W[a, b, i]$ is the minimum cost of covering all points in $X[a, b]$ that are above C_i , using only downward shadows of curves less wide than C_i . If such a covering does not exist, $W[a, b, i] = \infty$. For simplicity, we assume that C_N , the widest curve, contains no points in its downward shadow (that is, $X \cap S_N$ is empty). Our goal is then to determine $W[-\infty, \infty, N]$ via dynamic programming; the key structural result we need is the following:

Lemma 4.1.5. *If $X[a, b, i]$ is non-empty, then*

$$W[a, b, i] = \min \left\{ \min_{c \in (a, b)} \{W[a, c, i] + W[c, b, i]\}, \right. \\ \left. \min_{j < i} \{W[a, b, j] + w_j\} \right\}.$$

Proof. Clearly $W[a, b, i] \leq W[a, c, i] + W[c, b, i]$ for all $c \in (a, b)$. Also, for $j < i$, $W[a, b, j] + w_j$ is the cost of purchasing S_j and then covering the remaining points in $X[a, b]$ using regions less wide than S_j (and hence less wide than S_i). Thus $W[a, b, j] + w_j$ is a cost of a feasible solution to $(X[a, b, i], \mathcal{S}_{<i})$ and hence is at least $W[a, b, i]$. It follows that $W[a, b, i]$ is bounded above by the right hand side.

To show that $W[a, b, i]$ is bounded below by the right hand side, we let $\mathcal{Z} \subseteq \mathcal{S}_{<i}$ be a feasible set cover for $(X[a, b, i], \mathcal{S}_{<i})$. We consider two cases:

4. POLYNOMIAL-TIME ALGORITHMS FOR GEOMETRIC SET COVER

Case 1: There is some $c \in (a, b)$ such that $(c, C_i(c))$ is not covered by \mathcal{Z} . Let $\mathcal{Z}_{<c}$ be the set of all regions in \mathcal{Z} containing a point in $X[a, c, i]$, and let $\mathcal{Z}_{>c}$ be the set of all regions in \mathcal{Z} containing a point in $X[c, b, i]$. Let $Z \in \mathcal{Z}$. Since $Z \prec S_i$, by Proposition 4.1.4, $Z \setminus S_i$ is connected and thus cannot contain points both in $X[a, c, i]$ and $X[c, b, i]$. Hence $\mathcal{Z}_{<c} \cap \mathcal{Z}_{>c} = \emptyset$ and thus the cost of \mathcal{Z} is at least $W[a, c, i] + W[c, b, i]$.

Case 2: For all $c \in (a, b)$, the point $(c, C_i(c))$ is covered by \mathcal{Z} . Then \mathcal{Z} covers $X[a, b, i] \cup S_i$ and hence covers all points in $X[a, b]$. Let C_j be the widest curve in \mathcal{Z} , noting that $j < i$. Then the cost of \mathcal{Z} is at least $w_j + W[a, b, j]$ since $\mathcal{Z} \setminus S_j$ must cover all points in $X[a, b, j]$.

It follows that \mathcal{Z} must cost as much as either $\min_{c \in (a, b)} \{W[a, c, i] + W[c, b, i]\}$ or $\min_{j < i} \{W[a, b, j] + w_j\}$, and the result follows. \square

We can now prove our main result:

Proof of Theorem 4.1.3. Lemma 4.1.5 immediately implies the existence of a dynamic programming algorithm to compute $W[-\infty, \infty, N]$ and return a cover having that cost. It suffices to simply compute $W[a, b, i]$ recursively for all combinatorially relevant values of a , b , and i , using the min-cost covering of subproblems containing 1 or fewer points as the base case. There are at most $M + 1$ combinatorially relevant values of a and b when computing optimal costs $W[a, b, i]$ for subproblems, so there are $O(NM^2)$ distinct values of $W[a, b, i]$ to compute. Recursively computing $W[a, b, i]$ requires $O(M + N)$ table lookups, so the total runtime of our algorithm is $O(NM^2(M + N))$ primitive operations. \square

4.2 Applications: Half-planes, Pseudodisks, and More

A first application of our algorithm is directly to \mathbb{R}^2 -3-Sided-Box-SC. This problem is essentially equivalent to covering with shadows of horizontal line segments. With sufficient perturbation to forbid two 3-sided rectangles from being tangent in any way, the boundaries of such regions are clearly 2-intersecting. Additionally, the regions are downward shadows of x -monotone curves. We therefore obtain the following:

Theorem 4.2.1. *Weighted \mathbb{R}^2 -3-Sided-Box-SC admits an exact solution in polynomial time.*

Our next application shows us that the x -monotone requirement of Theorem 4.1.3 can be dropped if all the shadows contain the points $(x, -\infty)$ for all x . We express this idea in its *polar* form rather for more clarity.

Definition 4.2.2. A *configuration of pseudodisks* is a set of closed Jordan curves that intersect at most twice pairwise.

COVERING PROBLEM: \mathbb{R}^2 -**Origin-Containing-Pseudodisk-SC**
 ELEMENTS: A finite subset of \mathbb{R}^2
 SETS: A configuration of pseudodisks in \mathbb{R}^2 , each of which contains the origin (in other words, all of the pseudodisks share a common point, which may possibly lie on one or more of their boundaries).

Corollary 4.2.3. *There exists a polynomial-time exact algorithm for weighted \mathbb{R}^2 -**Origin-Containing-Pseudodisk-SC**. Furthermore, it runs in $O(NM^2(M+N))$ time on a set system consisting of M points and N pseudodisks.*

Proof. We refer the reader to Lemma 2.11 of [ANP⁺04], which shows us how to use a topological sweep curve method to transform the arrangement of pseudodisks into a topologically equivalent arrangement where all the pseudodisks are star-shaped about the origin. By examining the proof of this lemma, we note that this transformation can be accomplished in polynomial time since, with N pseudodisks, the sweep curve must be advanced at most $2\binom{N}{2}$ times. We then apply a standard polar-to-cartesian projective transformation about the origin (sending each point (x, y) to $(\text{atan2}(y, x), \sqrt{x^2 + y^2})$), which maps each star-shaped pseudodisk to the downward shadow of a positive valued x -monotone function on $[0, 2\pi)$, noting that these functions are pairwise 2-intersecting. Additionally, we map each point from the original set system into the appropriate cell of the transformed arrangement to obtain a topologically identical set system involving downward shadows of pairwise 2-intersecting x -monotone curves in \mathbb{R}^2 . It then suffices to apply Theorem 4.1.3. \square

A final application is to \mathbb{R}^2 -**Halfplane-SC**—covering with arbitrary (not necessarily lower) half-planes in \mathbb{R}^2 :

Corollary 4.2.4. *There are polynomial-time exact algorithms for weighted \mathbb{R}^2 -**Halfplane-SC** and \mathbb{R}^2 -**Halfplane-HS**, also running in $O(NM^2(M+N))$ time.*

Proof. We first apply an inversive transformation [Cox69] about the origin to map each half-plane to a disk whose boundary intersects the origin (and each point to its corresponding location after inversion). We shall obtain a new set system whose element-set incidence relation is unchanged, but is now a configuration of disks each containing the origin. These disks are pseudodisks, and thus the previous corollary can be applied to achieve the desired result for \mathbb{R}^2 -**Halfplane-SC**.

To obtain the result for \mathbb{R}^2 -**Halfplane-HS**, it suffices to observe that \mathbb{R}^2 -**Halfplane-SC** and \mathbb{R}^2 -**Halfplane-HS** are isomorphic problems due to projective duality [dBCvKO08]. \square

In the above application, we are giving an immediate and direct improvement over the dynamic programming algorithm of Har-Peled and Lee for weighted cover of points in the plane by half-planes [HPL08]; their method runs in $O(n^5)$ time on an instance with n points and half-planes. Our algorithm both generalizes theirs and reduces the run time by a factor of n .

4.3 Algorithm for a Hitting Set Problem

The hitting set version \mathbb{R}^2 -**3-Sided-Box-HS** of the 3-sided rectangle cover problem admits a much easier polynomial-time exact algorithm than the one used to solve \mathbb{R}^2 -**2-Intersecting-Shadow-SC**:

Theorem 4.3.1. *There exists a polynomial-time exact algorithm for weighted \mathbb{R}^2 -3-Sided-Box-HS. Moreover, it runs in $O(\min(N, M)^3 + M + N)$ time on a set system containing N points and M 3-sided rectangles.*

Proof. Suppose we are given a \mathbb{R}^2 -**3-Sided-Box-HS** set system consisting of a set X of N points and a family \mathcal{S} of M 3-sided rectangles (note that the roles of M and N have switched since we are now dealing with a hitting set problem). Define $\mathcal{S}[a, b]$ to be the family of all rectangles lying entirely inside $(a, b) \times (-\infty, \infty)$, and define $W[a, b]$ to be the minimum cost of hitting all rectangles in $\mathcal{S}[a, b]$. Define $y[a, b]$ to be the minimum y -coordinate of a top edge of a 3-sided rectangle in $\mathcal{S}[a, b]$, or ∞ if $\mathcal{S}[a, b]$ is empty. Finally, define $X[a, b]$ to be the set of all $x \in X$ lying in $(a, b) \times (-\infty, y]$.

Whenever $\mathcal{S}[a, b]$ is non-empty, hitting all rectangles in $\mathcal{S}[a, b]$ requires choosing at least one point in $X[a, b]$ (otherwise the lowest rectangle in $\mathcal{S}[a, b]$ will not be hit). However, any $x = (x_1, x_2) \in X[a, b]$ will hit all rectangles in $\mathcal{S}[a, b]$ whose horizontal range contains x_1 , since x is lower than the top edge of all rectangles in $\mathcal{S}[a, b]$. After choosing such an x , it thus only remains to cover those rectangles in $\mathcal{S}[a, x_1]$ and $\mathcal{S}[x_1, b]$. Via this argument, the following recurrence is immediate:

Claim 4.3.2. $W[a, b] = \min_{x=(x_1, x_2) \in X[a, b]} \{W[a, x_1] + W[x_1, b] + w_x\}$

Via this claim, it is simple to implement a dynamic programming algorithm that computes $W[-\infty, \infty]$ and returns a hitting set having that cost. There are at most $\min\{M, N\}$ combinatorially relevant values of a and b for which subproblems must be computed, and computing each subproblem requires at most $\min\{M, N\}$ table lookups because for any given a and b , the set $X[a, b]$ may contain at most $\min\{M, N\}$ combinatorially distinct points. The result follows. \square

Unfortunately, generalizing this result to other objects, such as downward shadows of parabolas in the plane, seems difficult.

Chapter 5

APX-Hardness Results

In this chapter, we shall present **APX**-hardness results for a variety of covering problems, most of which are geometric. All of these results stem from the **APX**-hardness of a new problem known as **Special-3SC**, which we believe may be of independent interest in proving the **APX**-hardness of other problems. Most of these results, including the **APX**-hardness of **Special-3SC** itself, have appeared before in [CG11].

5.1 Special-3SC

Historically, one of the first problems known not to admit a PTAS was **Min-Vertex-Cover**, and even in 1991, it was known that this hardness held even if the degree of the graph was at most 4 [PY91]. In 1997, an improvement was made by Alimonti and Kann, who showed that vertex cover remained **APX**-hard even when the resulting graph was 3-regular [AK00]. The resulting problem—**3-Regular-Graph-SC**—remained, for years, one of those widely used problems to reduce from when proving the **APX**-hardness of covering problems. Since vertex cover on graphs of degree at most 2 is solvable in polynomial time, it stands to reason that it should not be possible to obtain a stronger or more useful hardness result than the **APX**-hardness result of **3-Regular-Graph-SC**. However, we shall show that by imposing even more additional structure on an instance of **3-Regular-Graph-SC**, it is possible to preserve **APX**-hardness while introducing nice properties into the set system, which will allow us to obtain hardness results for other problems much more easily. The problem we obtain, which we call **Special-3SC**, has proven to be extremely versatile when attempting to prove the **APX**-hardness of geometric covering problems. We are not aware of anything else like it.

The formal definition of **Special-3SC** is as follows:

Definition 5.1.1. In an instance of **Special-3SC**, we are given a universe $U = AUWUXUYUZ$ comprising disjoint sets $A = \{a_1, \dots, a_n\}$, $W = \{w_1, \dots, w_m\}$, $X = \{x_1, \dots, x_m\}$, $Y = \{y_1, \dots, y_m\}$, and $Z = \{z_1, \dots, z_m\}$ where $2n = 3m$.

5. APX-HARDNESS RESULTS

We are also given a family \mathcal{S} of $5m$ subsets of U satisfying the following two conditions:

- For each $1 \leq t \leq m$, there are integers $1 \leq i < j < k \leq n$ such that \mathcal{S} contains the sets $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$, and $\{a_k, z_t\}$ (summing over all t gives the $5m$ sets contained in \mathcal{S}).
- For all $1 \leq t \leq n$, the element a_t is in exactly two sets in \mathcal{S} .

Our key result is the following:

Lemma 5.1.2. *Special-3SC is APX-complete.*

Proof. It is easy to see that **Special-3SC** is in **APX** because it is a covering problem of bounded frequency 2 and thus admits a greedy 2-approximation. To show that it is **APX-hard**, we shall exhibit an L-reduction from **3-Regular-Graph-SC** to **Special-3SC**, which is sufficient because **3-Regular-Graph-SC** is **APX-hard** [AK00], and L-reductions preserve **APX-hardness** [PY91].

We recall that a pair of functions (f, g) is an L-reduction from a minimization problem A to a minimization problem B if there are positive constants α and β such that for each instance x of A , the following hold:

- (L1) The function f maps instances of A to instances of B such that $\text{OPT}(f(x)) \leq \alpha \cdot \text{OPT}(x)$.
- (L2) The function g maps feasible solutions of $f(x)$ to feasible solutions of x such that $c_x(g(y)) - \text{OPT}(x) \leq \beta \cdot (c_{f(x)}(y) - \text{OPT}(f(x)))$, where c_x and $c_{f(x)}$ are the cost functions of the instances x and $f(x)$ respectively.

Given an instance x of **3-Regular-Graph-SC** on edges $\{e_1, \dots, e_n\}$ with vertices $\{v_1, \dots, v_m\}$ where $3m = 2n$, we define $f(x)$ be the **Special-3SC** instance containing the sets $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$, and $\{a_k, z_t\}$ for each 4-tuple (t, i, j, k) such that v_t is a vertex incident to edges e_i, e_j , and e_k with $i < j < k$. To define g , we suppose we are given a solution y to the **Special-3SC** instance $f(x)$. We take vertex v_t in our solution $g(y)$ of the **3-Regular-Graph-SC** instance x if and only if at least one of $\{a_i, w_t\}$, $\{a_j, x_t, y_t\}$, or $\{a_k, z_t\}$ is taken in y . We observe that g maps feasible solutions of $f(x)$ to feasible solutions of x since e_i is covered in $g(y)$ whenever a_i is covered in y .

Our key observation is the following:

Claim 5.1.3. $\text{OPT}(f(x)) = \text{OPT}(x) + 2m$.

Proof. For $1 \leq t \leq m$, we define the sets $\mathcal{P}_t = \{\{w_t, x_t\}, \{y_t, z_t\}\}$ and $\mathcal{Q}_t = \{\{a_i, w_t\}, \{a_j, x_t, y_t\}, \{a_k, z_t\}\}$. Call a solution \mathcal{C} of $f(x)$ *segregated* if for all $1 \leq t \leq m$, \mathcal{C} either contains all sets in \mathcal{P}_t and no sets in \mathcal{Q}_t , or contains all sets in \mathcal{Q}_t and no sets in \mathcal{P}_t .

Via local interchanging, we observe that there exists an optimal solution to $f(x)$ that is segregated. Specifically, when given an arbitrary optimal solution

\mathcal{C}^* of $f(x)$, we can construct a new solution \mathcal{C}' if, for each t , we simply take all sets in \mathcal{Q}_t whenever \mathcal{C}^* contains at least one set in \mathcal{Q}_t and otherwise take all sets in \mathcal{P}_t . It follows immediately that \mathcal{C}' is feasible if \mathcal{C}^* is, and it is not hard to see that the cost of \mathcal{C}' cannot exceed that of \mathcal{C}^* .

Additionally, our function g , when restricted to segregated solutions of $f(x)$, forms a bijection between them and feasible solutions of x . We check that g maps segregated solutions of size $2m + k$ to solutions of x having cost precisely k , and the claim follows. \square

Claim 5.1.3 implies that f satisfies property (L1) with $\alpha = 5$, since $\text{OPT}(x) \geq \frac{m}{2}$. Moreover, $c_x(g(y)) + 2m \leq c_{f(x)}(y)$ since both $\{w_t, x_t\}$ and $\{y_t, z_t\}$ must be taken in y whenever v_t is not taken in $g(y)$, and at least three of the sets in $\{\{a_i, w_t\}, \{w_t, x_t\}, \{a_j, x_t, y_t\}, \{y_t, z_t\}, \{a_k, z_t\}\}$ must be taken in y whenever v_t is taken in $g(y)$. Together with Proposition 5.1.3, this proves that g satisfies property (L2) with $\beta = 1$, completing the proof that (f, g) is an L-reduction. \square

5.2 Encodings of Special-3SC via Geometric Set Cover

We now use **Special-3SC** to show the **APX**-hardness of over a dozen geometric covering problems. Our method is, for a class \mathbf{C} of set systems, to prove that **Special-3SC** $\subseteq \mathbf{C}$, thereby showing that \mathbf{C} inherits the hardness of **Special-3SC**.

Theorem 5.2.1. *Unweighted geometric set cover is **APX**-hard with each of the following classes of geometric objects:*

- (C1) *Axis-aligned rectangles in \mathbb{R}^2 , even when all rectangles have lower-left corner in $[-1, -1 + \epsilon] \times [-1, -1 + \epsilon]$ and upper-right corner in $[1, 1 + \epsilon] \times [1, 1 + \epsilon]$ for an arbitrarily small $\epsilon > 0$.*
- (C2) *Axis-aligned ellipses in \mathbb{R}^2 , even when all ellipses have centers in $[0, \epsilon] \times [0, \epsilon]$ and major and minor axes of length in $[1, 1 + \epsilon]$.*
- (C3) *Axis-aligned slabs in \mathbb{R}^2 , each of the form $[a_i, b_i] \times [-\infty, \infty]$ or $[-\infty, \infty] \times [a_i, b_i]$.*
- (C4) *Axis-aligned rectangles in \mathbb{R}^2 , even when the boundaries of each pair of rectangles intersect exactly zero times or four times.*
- (C5) *Downward shadows of line segments in \mathbb{R}^2 .*
- (C6) *Downward shadows of (graphs of) univariate cubic functions in \mathbb{R}^2 .*
- (C7) *Unit balls in \mathbb{R}^3 , even when all the balls contain a common point.*
- (C8) *Axis-aligned cubes in \mathbb{R}^3 , even when all the cubes contain a common point and are of similar size.*

5. APX-HARDNESS RESULTS

(C9) *Half-spaces in \mathbb{R}^4 .*

(C10) *Fat wedges in \mathbb{R}^2 , each of which has an opening angle in $[\pi - \epsilon, \pi]$.*

Additionally, unweighted geometric hitting set is APX-hard with each of the following classes of objects:

(H1) *Axis-aligned slabs in \mathbb{R}^2 .*

(H2) *Axis-aligned rectangles in \mathbb{R}^2 , even when the boundaries of each pair of rectangles intersect exactly zero times or four times.*

(H3) *Unit balls in \mathbb{R}^3 .*

(H4) *Half-spaces in \mathbb{R}^4 .*

Before we proceed to the encodings that prove this theorem, we make a few comments about the implications of our results.

Mustafa and Ray ask if their local improvement approach outlined in [MR10] might yield a PTAS for a wider class of instances; Theorem 5.2.1 immediately rules this out for all of the covering and hitting set problems listed above by proving that no PTAS exists for them unless $P = NP$. Item (C1) demonstrates that even tiny perturbations can destroy the behaviour of the local search method. (C2) rules out the possibility of a PTAS for arbitrarily fat ellipses (that is, ellipses that are within ϵ of being perfect circles). (C5) and (C6) stand in contrast to our algorithm from Chapter 4, which proves that geometric set cover is polynomial-time solvable when the objects are downward shadows of horizontal line segments or quadratic functions. In the case of (C4) and (H2), the intersection graph of the rectangles is a comparability graph (and hence a perfect graph); even then, neither set cover nor hitting set admits a PTAS. (C7), (C8), (C9), (H3), and (H4) complement the result of Mustafa and Ray by showing that their algorithm fails in higher dimensions. (C10) stands in contrast to the fact that covering with half-planes is exactly solvable in polynomial time.

A natural gap exists between our results (C6), (C7), and (H3) and the results of Mustafa and Ray. For unit balls in \mathbb{R}^3 , the existence of a PTAS remains an open question for both set cover and hitting set. Additionally, set cover with arbitrary disks or squares in \mathbb{R}^2 remains open (the result of Mustafa and Ray can only provide a PTAS for the hitting set version, or the covering version when all disks or squares have the same radius).

The beauty of **Special-3SC** is that it allows many of our geometric **APX**-hardness results to follow immediately from simple “proofs by pictures” (see Figure 5.2). The key property of **Special-3SC** is that we can divide the elements into two sets A and $B = W \cup X \cup Y \cup Z$, and linearly order B in such a way that all sets in \mathcal{S} are either two adjacent elements from B , one from B and one from A , or two adjacent elements from B and one from A . We need only make $[w_t, x_t, y_t, z_t]$ appear consecutively in the ordering of B . Knowing this, we present a proof of Theorem 5.2.1:

Proof of Theorem 5.2.1.

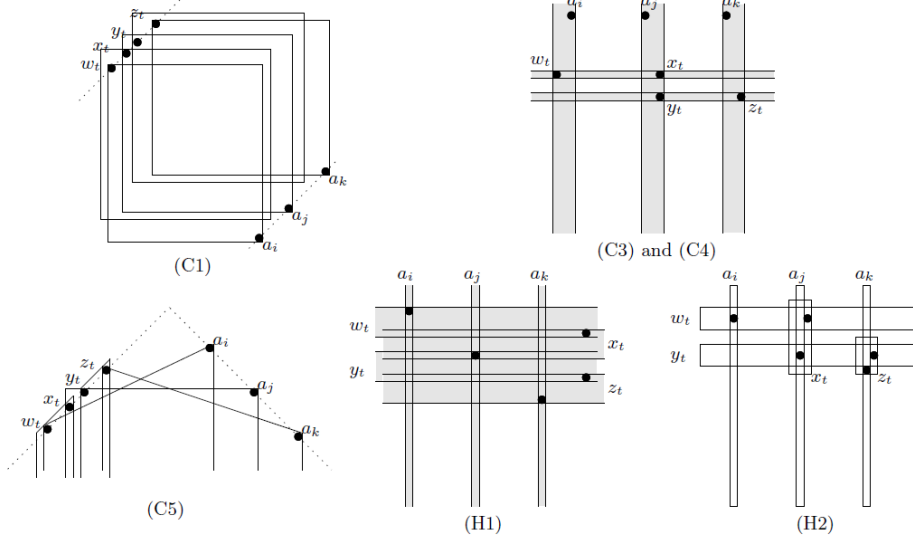


Figure 5.1: APX-hardness proofs of geometric set cover problems.

For (C1), we simply place the elements of A on the line segment $\{(x, x-2) : x \in [1, 1+\epsilon]\}$ and place the elements of B , in order, on the line segment $\{(x, x+2) : x \in [-1, -1+\epsilon]\}$, for a sufficiently small $\epsilon > 0$. As we can see immediately from Figure 5.2, each set in \mathcal{S} can be encoded as a fat rectangle in the class (C1).

(C2) is similar. It is not difficult to check that each set can be encoded as a fat ellipse in this class.

For (C3), we place the elements of A on a horizontal line (the top row). For each $1 \leq t \leq m$, we create a new row containing $\{w_t, x_t\}$ and another containing $\{y_t, z_t\}$ as shown in Figure 5.2. This time, we will need the second property in Definition 5.1.1—that each a_i appears in two sets. If $\{a_i, w_t\}$ is the first set that a_i appears in, we place w_t slightly to the left of a_i ; if it is the second set instead, we place w_t slightly to the right of a_i . Similarly, the placement of x_t, y_t (resp. w_t) depends on whether a set of the form $\{a_j, x_t, y_t\}$ (resp. $\{a_k, w_t\}$) is the first or second set that a_j (resp. a_k) appears in. As we can see from Figure 5.2, each set in \mathcal{S} can be encoded as a thin vertical or horizontal slab.

(C4) is similar to (C3), with the slabs replaced by thin rectangles. For example, if $\{a_i, w_t\}$ and $\{a_i, w_{t'}\}$ are the two sets that a_i appears in, with w_t located higher than $w_{t'}$, we can make the rectangle for $\{a_i, w_t\}$ slightly wider than the rectangle for $\{a_i, w_{t'}\}$ to ensure that these two rectangles intersect 4 times.

For (C5), we can place the elements of A on the ray $\{(x, -x) : x > 0\}$ and the elements of B , in order, on the ray $\{(x, x) : x < 0\}$. The sets in \mathcal{S} can be

5. APX-HARDNESS RESULTS

encoded as downward shadows of line segments as in Figure 5.2.

(C6) is similar to (C5). One way is to place the elements of A on the line segment $\ell_A = \{(x, x) : x \in [-1, -1 + \epsilon]\}$ and the elements of B (in order) on the line segment $\ell_B = \{(x, 0) : x \in [1.5, 1.5 + \epsilon]\}$. For any $a \in [-1, -1 + \epsilon]$ and $b \in [1.5, 1.5 + \epsilon]$, the cubic function $f(x) = (x - b)^2[(a + b)x - 2a^2]/(b - a)^3$ is tangent to ℓ_A and ℓ_B at $x = a$ and $x = b$. (The function intersects $y = 0$ also at $x = 2a^2/(a + b) \gg 1.5 + \epsilon$, far to the right of ℓ_B). Thus, the size-2 sets in \mathcal{S} can be encoded as cubics. A size-3 set $\{a_j, x_t, y_t\}$ can also be encoded if we take a cubic with tangents at a_j and x_t , shift it upward slightly, and make x_t and y_t sufficiently close.

For (C7), we place the elements in A on a circular arc $\gamma_A = \{(x, y, 0) : x^2 + y^2 \leq 1, x, y \geq 0\}$ and the elements in B (in order) on the vertical line segment $\ell_B = \{(0, 0, z) : z \in [1 - 2\epsilon, 1 - \epsilon]\}$. (This idea is inspired by a known construction [BD99], after much simplification). We can ensure that every two points in A have distance $\Omega(\sqrt{\epsilon})$ if $\epsilon \ll 1/n^2$. The technical lemma below allows us to encode all size-2 sets (by setting $b = b'$) and size-3 sets by unit balls containing a common point.

Lemma 5.2.2. *Given any $a \in \gamma_A$ and $b, b' \in \ell_B$, there exists a unit ball that (i) intersects γ_A at an arc containing a of angle $O(\sqrt{\epsilon})$, (ii) intersects ℓ_B at precisely the segment from b to b' , and (iii) contains $(1/\sqrt{2}, 1/\sqrt{2}, 1)$.*

Proof. Say $a = (x, y, 0)$, $b = (0, 0, z - h)$, $b' = (0, 0, z + h)$. Consider the unit ball K centered at $c = ((1 - h^2)x, (1 - h^2)y, z)$. Then (ii) is self-evident and (iii) is straightforward to check. For (i), note that a lies in K since $\|a - c\|^2 = h^2 + z^2 \leq \epsilon^2 + (1 - \epsilon)^2 < 1$. On the other hand, if a point $p \in \gamma_A$ lies in the unit ball, then letting $a' = ((1 - h^2)x, (1 - h^2)y, 0)$, we have $\|p - c\|^2 = \|p - a'\|^2 + z^2 \leq 1$, implying $\|p - a\| \leq \|p - a'\| + \|a' - a\| \leq \sqrt{1 - z^2} + h = O(\sqrt{\epsilon})$. \square

(C8) is similar to (C1); we place the elements in A on the line segment $\ell_A = \{(t, t, 0) : t \in (0, 1)\}$ and the elements in B on the line segment $\ell_B = \{(0, 3 - t, t) : t \in (0, 1)\}$. For any $(a, a, 0) \in \ell_A$ and $(0, 3 - b, b) \in \ell_B$, the cube $[-3 + b + 2a, a] \times [a, 3 - b] \times [-3 + a + 2b, b]$ is tangent to ℓ_A at $(a, a, 0)$, is tangent to ℓ_B at $(0, 3 - b, b)$, and contains $(0, 1, 0)$. Size-3 sets $\{a_j, x_t, y_t\}$ can be encoded by taking a cube with tangents at a_j and x_t , expanding it slightly, and making x_t and y_t sufficiently close. We can move the lines further apart to make the cubes within epsilon of being unit cubes.

(C9) follows from (C7) by the standard lifting transformation that maps points $(x, y, z) \in \mathcal{R}^3$ to $(x, y, z, x^2 + y^2 + z^2) \in \mathcal{R}^4$ and balls $\{(x, y, z) : (x - a)^2 + (y - b)^2 + (z - c)^2 \leq r^2\}$ to half-spaces $\{(x, y, z, w) : w - 2ax - 2by + 2cz \leq r^2 - a^2 - b^2 - c^2\}$ [dBCvKO08].

For (C10), we shall place the elements of A and B respectively along two circular arcs $\ell_A = \{(\cos t, \sin t) : t \in (0, \epsilon)\}$, and $\ell_B = \{(\cos t, 2 - \sin t) : t \in (0, \epsilon)\}$. Any point on either arc defines a unique tangent to that arc, and thus any pair of points, one on ℓ_A and one on ℓ_B , define a pair of tangents that meet at a unique intersection point (which will be somewhat close to $(1, 1)$). It follows that a wedge of opening angle close to π can be chosen to be tangent to

any point on ℓ_A and any point on ℓ_B . Again, for the size-3 sets, it suffices to expand each wedge slightly and make x_t and y_t sufficiently close.

For (H1), we map each element a_i to a thin vertical slab. For each $1 \leq t \leq m$, we map $\{w_t, x_t, y_t, z_t\}$ to a cluster of four thin horizontal slabs as in Figure 5.2. Each set in \mathcal{S} can be encoded as a point in the arrangement.

(H2) is similar; see Figure 5.2.

(H3) follows from (C7) by duality; see Proposition 2.3.16.

(H4) follows from (C9) by duality; see Proposition 2.3.18. \square

5.3 Hardness of Priority Tree Cover

Here, we prove that **Priority-Vertical-Tree-SC**—the priority version of the totally unimodular problem **Vertical-Tree-SC**—is **APX**-hard. An alternate proof that **Priority-Tree-SC** is **APX**-hard can be found in [CGK10a], but our new proof, which uses **Special-3SC**, is significantly shorter.

As with the previous results, we will prove that **Priority-Vertical-Tree-SC** is **APX**-hard by establishing that it can encode **Special-3SC**:

Theorem 5.3.1. ***Special-3SC** \subseteq **Priority-Vertical-Tree-SC**, and therefore **Priority-Vertical-Tree-SC** is **APX**-hard.*

Proof. Suppose we are given an instance of **Special-3SC** with parameters m and n as usual. Let P be a path containing precisely $4m$ edges, and write the edges of P , in order, as

$$P = [e_{w_1}, e_{x_1}, e_{y_1}, e_{z_1}, e_{w_2}, e_{x_2}, e_{y_2}, e_{z_2}, \dots, e_{z_m}],$$

identifying them with elements from our **Special-3SC** instance. Let u be one of the end vertices of P , and let S be a star rooted at a vertex v having n neighbours (that is, v is adjacent to n other vertices, none of which are adjacent to one another). We construct a tree T by identifying u with v , creating a caterpillar graph comprising a single path of length $4m$ leading to a vertex v of degree $n + 1$, adjacent to the path and n leaves. We assign priorities to the edges of T such that the priority of an edge in P is a decreasing function of its distance from v , and the priority of all edges in S is zero. We identify the edges of S with elements a_1 through a_n of our original **Special-3SC** instance. It is then not hard to see that a path in T , of the correctly chosen supply priority, can cover exactly 2 adjacent elements in P and one element of S . The result follows. \square

Chapter 6

Improved Quasi-Uniform Sampling for Weighted Set Cover

Here, we present our quasi-uniform sampling algorithm for weighted set cover. We build off of the ideas in Varadarajan’s [Var10] result, extending the scope of his technique to more general, not necessarily geometric, set cover instances. Additionally, we improve the approximation guarantee in some cases as a result of modifications to both the algorithm and its analysis.

The other important feature of our algorithm is that its approximation factor varies according to an instance-specific parameter known as *shallow cell complexity* (SCC). This is a purely combinatorial property of set cover instances, and checking that a set system has low SCC can be done much more easily than, e.g., establishing bounds on union complexity for a non-geometric set system. In fact, using SCC, we recover all of the previous results of Varadarajan [Var10] as well as generalize the results of Clarkson and Varadarajan and all related sequels [CV07, Var09, AES10] to the weighted case, using no additional effort. Moreover, we also obtain many new results for various other weighted covering problems, including non-geometric ones.

All of the results in this section are from [CGKS12].

6.1 Shallow Cell Complexity

One of the key technical concepts used by Varadarajan [Var10] is the *cell complexity* of a given configuration of geometric objects. Informally, in a configuration of objects in Euclidean space, a *cell* is a maximal connected region consisting of points that all lie within precisely the same set of objects. The *depth* of a cell is the number of objects that define the cell. In his algorithm, Varadarajan uses an earlier result by Clarkson and Shor [CS89] that shows that

geometric set cover instances with low union complexity have a small number of cells of large depth.

For our purposes, we shall strip away the topological details underpinning the formal geometric definition of cells, leaving us with a purely combinatorial notion of ‘cell’ for the matrix world. We call two rows A_i and A_j of a 0, 1-matrix A *equivalent* if they contain ones in precisely the same columns. The *cells* of A are then defined to be the resulting equivalence classes, and the *depth* of a cell is the number of ones in any one of its rows. For example, the first and third row of matrix A on the right are equivalent and have depth two. There are two more cells formed by rows two and four, of depth two and one, respectively. We define the key property used in our algorithm:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The *cells* of A are then defined to be the resulting equivalence classes, and the *depth* of a cell is the number of ones in any one of its rows. For example, the first and third row of matrix A on the right are equivalent and have depth two. There are two more cells formed by rows two and four, of depth two and one, respectively. We define the key property used in our algorithm:

Definition 6.1.1 (Cell Complexity). Let $f(n, k)$ be a function that is non-decreasing in both n and k . A binary matrix A with N columns has *shallow cell complexity* (SCC) f if for all $1 \leq k \leq n \leq N$ and for all sub-matrices A^* of A containing exactly n columns, the number of cells of A^* of depth k or fewer is at most $f(n, k)$. A class of set systems \mathbf{C} has SCC f if and only if all $A \in \mathbf{C}$ do.

Intuitively, matrices with low SCC are those for whom all submatrices have few distinct rows containing few ones. The SCC of a matrix is, in some sense, the worst case density of any cluster of such rows in a submatrix of A . We note that SCC is already a *hereditary* property; a matrix’s SCC is an upper bound for the SCC of all of its submatrices.

Below are some examples of SCC bounds for various simple classes of set system matrices:

- General binary matrices have SCC at most $\binom{n}{k}$.
- Binary matrices that do not contain the submatrix $[0, 1]$ have SCC at most $k + 1$.
- A set system matrix has SCC at most $O(n^d)$ whenever its underlying set system has VC codimension at most d (see [HW87]).
- Binary network matrices have SCC $O(n)$, as we show in Lemma 6.5.2.

6.2 The Quasi-Uniform Sampling Algorithm

Our main result shows that set cover instances with small SCC are well approximable:

Theorem 6.2.1. *Let $\phi(n)$ be a non-decreasing function of n , and let $c \geq 0$ be a constant. Suppose \mathbf{C} is a class of set cover instances with SCC $f(n, k) = n\phi(n)k^c$.*

6.2. THE QUASI-UNIFORM SAMPLING ALGORITHM

Then there exists a quasi-uniform ϵ -net finder of size $O(\frac{1}{\epsilon} \max\{1, \log \phi(N)\})$ for \mathbf{C} (where N is the number of columns in a given $A \in \mathbf{C}$).

Via Theorem 2.4.20, we can immediately obtain the following approximability result:

Corollary 6.2.2. *Let $\phi(n)$ be a non-decreasing function of n , and let $c \geq 0$ be a constant. Suppose \mathbf{C} is a class of set cover instances with SCC $f(n, k) = n\phi(n)k^c$. Then there is a randomized polynomial-time LP-relative $O(\max\{1, \log \phi(OPT)\})$ -approximation algorithm for the weighted set cover problem for \mathbf{C} .*

As mentioned, our algorithm follows Varadarajan's general sampling framework, ensuring quasi-uniformity [Var10]. Varadarajan's version of the algorithm and analysis gives an $O(\log \phi(n))$ -approximation when $\phi(n) \in \omega(\log^{(j)} n)$ for some constant j , but obtains only a $2^{O(\log^* n)}$ -approximation in the case of $\phi(n) = O(1)$. We provide a refined analysis that eliminates the $2^{O(\log^* n)}$ factor. At the same time, our algorithm also simplifies Varadarajan's in several ways. In Section 6.3 we also show how our algorithm can be derandomized.

We now describe our proof of Theorem 6.2.1. Recall that a set system matrix A is k -deep if all rows of A contain at least k ones. We shall present an ϵ -net finder obtaining the following: Our goal is to provide a polynomial algorithm to produce a small, quasi-uniform L/N -net of an L -deep $M \times N$ set system matrix A^* of SCC $f(n, k) = n\phi(n)k^c$. Throughout this section, we let $\ell(N) = \max\{1, \log \phi(N)\}$. For simplicity, we shall assume that $f(n, k)$ is known a priori, although this is unnecessary if standard binary search techniques are employed to guess c and $\ell(N)$.

The algorithm proceeds in a series of sequential phases. At the start of a phase, we are given a k -deep $m \times n$ submatrix A of A^* with SCC $f(n, k)$ as inherited from A^* . The eventual goal is to produce a set cover for the subproblem induced by A , but much of the work will be put off until future phases, which operate on increasingly small nested sub-matrices of A . The purpose of a single phase is simply to partition the columns of A into three categories:

- *Forced* columns: those that will definitely be taken in our set cover;
- *Rejected* columns: those that will definitely not be taken in our set cover;
- *Retained* columns: those for which the decision to force or reject will be deferred until a future phase of the algorithm.

Given such a partition, we define B to be the submatrix of A obtained by deleting all forced columns, rejected columns, and rows with a one in any forced column (rows *covered* by a forced column). The subsequent phase of the algorithm operates on B .

The algorithm we present reduces the size and depth of A by a factor of approximately $\frac{1}{2}$ during each phase. Reducing by $\frac{1}{2}$ is a somewhat arbitrary choice that we make to optimize the simplicity of our presentation. We contrast our algorithm with that of Varadarajan [Var10], who reduces the depth

of A from k to $\log k$ during each phase, picking up an unavoidable $2^{O(\log^* n)}$ in the approximation factor when the shallow cell complexity of A is very low. We avoid this by employing a forcing scheme that is somewhat simpler than Varadarajan’s original “forced addition” scheme [Var10].

Roughly speaking, in each phase, we randomly and independently mark each column of A with probability $\frac{1}{2} + H(N, k)$ for a carefully-chosen function $H(N, k)$. We then force some columns of A in order to cover rows that contain ones in fewer than $\frac{k}{2}$ marked columns, and retain the marked columns that remain. The exact manner in which we force will be described later; the key trick is to use low SCC to obtain a forcing rule in which no column is forced with high probability. Formally, our marking and forcing rules will achieve the following phase:

1. The output B is $\frac{k}{2}$ -deep (each row of B contains at least $\frac{k}{2}$ ones).
2. Each column of A is retained with probability at most $\frac{1}{2} + h(N, k)$.
3. Each column of A is forced with probability at most k^{-2} .

We will see later that it suffices to take $h(N, k) = O\left(\sqrt{\frac{\log k + \ell(N)}{k}}\right)$. Our analysis exploits the following key property of shallow cell complexity:

Lemma 6.2.3. *Suppose A is k -deep, has n columns, and has shallow cell complexity $f(n, k) = n\phi(n)k^c$. Then there is a column S of A such that the number of cells of depth exactly k that contain a one in S is at most $\phi(n)k^{c+1}$.*

Proof. Let A' be the submatrix of A obtained by deleting all rows that have more than k ones and eliminating duplicate copies of rows. Since A has shallow cell complexity f , so does A' and thus there are at most $n\phi(n)k^c$ rows in A' . Each row of A' contains exactly k ones, so A' contains at most $n\phi(n)k^{c+1}$ ones. Thus the average number of ones per column in A' is $\phi(n)k^{c+1}$ and it follows that some column of A' contains at most $\phi(n)k^{c+1}$ ones. The corresponding column of A then contains at most $\phi(n)k^{c+1}$ ones in cells of A having depth exactly k . \square

We now use this result to develop a procedure that takes a k -deep set system (X, \mathcal{R}) of low shallow cell complexity and returns a highly structured partition of X into *clusters* of points, each of which lies within the common intersection of some k sets in \mathcal{R} . We wish to assign each cluster to be the *responsibility* of some set containing it such that no set is responsible for too many clusters. In our full algorithm, we will force sets when the clusters they are responsible for are insufficiently covered, and the fact that no column is responsible for too many clusters will enable us to obtain an upper bound on the probability that a column is forced. We first give a definition:

Definition 6.2.4. In a set system matrix A , a group of rows R is called a *k -cluster* if there exists a set C of k columns of A that each contain a one in all rows of R . In such a case, C is said to *support* R .

For example, rows 2 and 4 of matrix A to the right form a 2-cluster supported by columns 2 and 3. The next result follows immediately:

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Proposition 6.2.5. *Let A be a set system matrix and let A' be a submatrix of A . Suppose R is a cell of depth k in A' (that is, a collection of identical rows in A' , each containing k ones). Then the rows in R form a k -cluster when regarded as rows of A (note that they are not necessarily identical in A).*

We now state our key lemma:

Lemma 6.2.6. *Suppose a set system matrix A is k -deep, has n columns, and has shallow cell complexity $f(n, k) = n\phi(n)k^c$. Denote by X and \mathcal{S} the rows and columns of A respectively. Then there exists a function $\gamma : X \rightarrow \mathcal{S}$ such that:*

- $\gamma(x)$ is a column containing a one in row x , and
- for each $S \in \mathcal{S}$, the pre-image $\gamma^{-1}(S) = \{x \in X : \gamma(x) = S\}$ can be partitioned into $\phi(n)k^{c+1}$ k -clusters of A .

Moreover, such a function γ can be computed in polynomial time.

Proof. We define an iterative procedure to assign $\gamma(x)$ for each $x \in X$ in polynomial time:

-
- 1: Initialize $A_1 \leftarrow A$ and $i \leftarrow 1$.
 - 2: Find a column $S_i \in \mathcal{S}$ such that at most $\phi(n)k^{c+1}$ cells of A_i having depth exactly k contain a one in S_i (one exists by Lemma 6.2.3). Let $Y_i \subseteq X$ be all rows of A_i that are members of the $\phi(n)k^{c+1}$ cells.
 - 3: Set $\gamma(x) = S_i$ for each row $x \in Y_i$.
 - 4: $A_{i+1} \leftarrow$ submatrix of A_i obtained by deleting the column S_i and all rows in Y_i from A_i .
 - 5: If A_{i+1} contains no rows, $\gamma(x)$ is defined for all rows $x \in X$ and we may terminate.
 - 6: Otherwise, increment i and go back to step 2.
-

We note that after the deletions in step 4, A_{i+1} is still k -deep because rows in $X \setminus Y_i$ either contain more than k ones or contain a zero in column S_i ; in either case, their depth after the deletion of column S_i cannot be less than k . Additionally, the shallow cell complexity of A is inherited by sub-matrices, so A_{i+1} always has shallow cell complexity $f(n, k) = n\phi(n)k^c$ throughout the procedure. This permits the application of Lemma 6.2.3 throughout the iterations of the procedure, implying that our procedure terminates and thus assigns a value to $\gamma(x)$ for each $x \in X$. Additionally, $\gamma(x)$ is always a column containing a one in row x , as we require.

6. IMPROVED QUASI-UNIFORM SAMPLING FOR WEIGHTED SET COVER

Additionally, the pre-image $Y_i = \gamma^{-1}(S_i)$ is a collection of at most $\phi(n)k^{c+1}$ cells of A_i . Proposition 6.2.5 then implies that $\gamma^{-1}(S_i)$ can be partitioned into at most $\phi(n)k^{c+1}$ k -clusters of A , completing the proof. \square

Formally, we say that S is *responsible* for x whenever $\gamma(x) = S$. With our key lemma in hand, we can finally provide a formal description of a phase:

-
- 1: **Input:** m by n set system matrix A of depth k with SCC $n\phi(n)k^c$
 - 2: **if** $\log k \geq \frac{k}{12(c+3)}$ **or** $\ell(N) \geq \frac{k}{12(c+3)}$ **then**
 - 3: Force every column of A and terminate
 - 4: **else**
 - 5: Mark each column of A independently with probability $\frac{1}{2} + h(N, k)$
 - 6: Obtain a function γ as described in Lemma 6.2.6
 - 7: **for all** rows x of A **do**
 - 8: **if** x does not contain at least $\frac{k}{2}$ ones in marked columns **then**
 - 9: Force $\gamma(x)$
 - 10: **end if**
 - 11: **end for**
 - 12: Reject the remaining columns of A that have been neither forced nor marked
 - 13: **end if**
 - 14: Obtain matrix B from A by deleting forced columns, rejected columns, and rows with a one in a forced column
 - 15: **Output:** B
-

It is clear that, after a single phase, B will be $\frac{k}{2}$ -deep, as any row x of A that does not lie in $\frac{k}{2}$ marked columns is deleted when $\gamma(x)$ is forced. It follows that each phase halves the depth of A , and thus the algorithm will terminate in at most $\lceil \log L \rceil$ phases when given a set system matrix of depth L ; hence, the algorithm runs in polynomial time.

We also verify that the final output does indeed form a set cover of A . Rows of A are deleted during the algorithm if and only if they contain a one in a forced column, and thus are covered. Rows that are never deleted are covered in the final phase when all columns of A are taken.

To ensure that our algorithm is feasible, we must verify that the marking probability $h(N, k) + \frac{1}{2}$ is at most one. This is easy from our choice of terminating condition. During non-terminating phases of the algorithm, we have both $\log k < \frac{k}{12(c+3)}$ and $\ell(N) < \frac{k}{12(c+3)}$, and thus:

$$\begin{aligned} \log k + \ell(N) &< \frac{k}{6(c+3)} \\ \implies \frac{3(c+3)}{2} \frac{\log k + \ell(N)}{k} &< \frac{1}{4} \\ \implies \sqrt{\frac{3(c+3)}{2} \frac{\log k + \ell(N)}{k}} + \frac{1}{2} &< 1. \end{aligned}$$

The final technical challenge remaining is to bound the probability that a column is forced at some point during the algorithm. We begin by bounding the probability that a single column is forced during a single phase:

Claim 6.2.7. *In a single non-terminating phase of the algorithm, each column S of A is forced with probability at most k^{-2} .*

Proof. We recall by Lemma 6.2.6 that the pre-image $\gamma^{-1}(S) = \{x \in X : \gamma(x) = S\}$ can be partitioned into $\phi(n)k^{c+1}$ k -clusters of A . Fix such a partition as obtained in Lemma 6.2.6. With this partition under consideration, suppose a row x of A has $\gamma(x) = S$ and lies in a k -cluster R supported by a set of columns C . A row in R can only cause S to be forced if fewer than $\frac{k}{2}$ columns in C are marked. We shall bound the probability of this happening in order to bound the probability that any row in R is insufficiently covered by the marked columns. We let Z be a random variable indicating the number of columns of C that are marked. Define

$$\mu = E[Z] = k \left[\frac{1}{2} + h(N, k) \right].$$

Then applying the Chernoff bound yields:

$$\begin{aligned} & \Pr \left[\text{Fewer than } \frac{k}{2} \text{ columns in } C \text{ marked} \right] \\ & \leq \Pr \left[Z \leq \frac{k}{2} \right] = \Pr \left[Z \leq \left(1 - \frac{kh(N, k)}{\frac{k}{2} + kh(N, k)} \right) \mu \right] \\ & \leq \text{Exp} \left(-\frac{1}{3} \left(\frac{kh(N, k)}{\frac{k}{2} + kh(N, k)} \right)^2 \left(\frac{k}{2} + kh(N, k) \right) \right) \\ & \leq \text{Exp} \left(-\frac{2}{3} kh(N, k)^2 \right). \end{aligned}$$

Since $\gamma^{-1}(S)$ can be partitioned into $\phi(n)k^{c+1}$ k -clusters of A , by the union bound, the probability that S is forced during an individual phase is at most

$$\phi(n)k^{c+1} \text{Exp} \left(-\frac{2}{3} kh(N, k)^2 \right).$$

Taking $h(N, k) = \sqrt{\frac{3}{2} \frac{(c+3) \log k + \ell(N)}{k}}$ and recalling that $n \leq N$ during all phases of the algorithm, this is at most

$$\phi(n)k^{c+1} \text{Exp} \left(-(c+3) \log k - \log(\phi(n)) \right) = k^{-2}. \quad \square$$

The previous claim essentially proves that our additional sampling probability $h(N, k)$ is *big enough* to cause the forcing probability in each phase to be relatively low. Our next claim shows us that $h(N, k)$ is still *small enough* for the probability of a column surviving t phases of sampling to decay exponentially in t . The function $h(N, k)$ is indeed quite finely tuned in order to exhibit both of these features.

Claim 6.2.8. *After $t \geq 1$ phases of our algorithm, the probability of any given column of the original M by N matrix A^* still remaining is $\frac{O(1)}{2^t}$.*

Proof. This is clearly true after the terminating condition has occurred. Before that happens, A is still at least $\frac{L}{2^t}$ -deep after i phases, so the probability that a row is retained during phase $i+1$ is at most $\frac{1}{2} + h(N, \frac{L}{2^i})$. Multiplying over all phases yields an upper bound of

$$P_t = \prod_{i=0}^{t-1} \left(\frac{1}{2} + h \left(N, \frac{L}{2^i} \right) \right) = \prod_{i=0}^{t-1} \left(\frac{1}{2} + O(1) \sqrt{\frac{(2^i)(\log L - i + \ell(N))}{L}} \right).$$

We recall that at a phase in which the initial depth is k , we must always have $\ell(N) < \frac{k}{12(c+3)}$, otherwise we terminate. If we have not yet terminated in phase t , we may take $k = \frac{L}{2^t}$, the depth after t phases, and obtain $\ell(N) < \frac{L}{12(c+3)2^t}$. Combining with the above yields:

$$\begin{aligned} P_t &\leq \prod_{i=0}^{t-1} \left(\frac{1}{2} + O(1) \sqrt{\frac{(2^i)(\log L - i)}{L}} + \frac{O(1)}{2^{t-i}} \right) \\ &= \frac{1}{2^t} \prod_{j=\log L-t}^{\log L} \left(1 + O(1) \sqrt{\frac{j}{2^j} + 2^{\log L-t-j}} \right) \\ &\leq \frac{O(1)}{2^t} \text{Exp} \left(\sum_{j=\log L-t}^{\log L} \left(\sqrt{\frac{j}{2^j} + 2^{\log L-t-j}} \right) \right) \\ &\leq \frac{O(1)}{2^t} \text{Exp}(O(1)) = \frac{O(1)}{2^t}. \end{aligned}$$

The final inequality follows from the fact that

$$\int_0^\infty \sqrt{\frac{x}{2^x}} dx = \frac{\sqrt{2\pi}}{(\ln 2)^{3/2}} \approx 4.34362 \in O(1). \quad \square$$

Claim 6.2.9. *A column is forced during the final phase of the algorithm with probability $O\left(\frac{\ell(N)}{L}\right)$, where $\ell(N) = \max\{1, \log \phi(N)\}$.*

Proof. The algorithm terminates when either $\log k \geq \frac{k}{12(c+3)}$ or $\ell(N) \geq \frac{k}{12(c+3)}$. In particular, during the terminating phase of the algorithm in which every column is forced, we must have $\log k + \ell(N) \geq \frac{k}{12(c+3)}$ from which it follows that $k \leq O(1)\ell(N)$. Thus at least $O(1) \log\left(\frac{L}{\ell(N)}\right)$ phases are required to reach the terminating condition. Consequently, taking $t = O(1) \log\left(\frac{L}{\ell(N)}\right)$ in the previous claim yields the desired result. \square

We finally prove that our algorithm returns a quasi-uniform cover:

Claim 6.2.10. *Throughout all phases of the algorithm, a given column is forced with probability at most $O\left(\frac{\ell(n)}{L}\right)$, where $\ell(n) = \max\{1, \log \phi(n)\}$.*

Proof. By the previous claim, it is sufficient to obtain the stated bound for non-terminating phases of the algorithm. After phase i , the depth is $\frac{L}{2^i}$ and thus the probability of a column both remaining after i phases and being forced during phase $i + 1$ is

$$\frac{O(1)}{2^i} \left(\frac{L}{2^i}\right)^{-2} = \frac{O(1)}{L} \frac{2^i}{L}.$$

Summing over all phases yields a bound of at most

$$\frac{O(1)}{L} \sum_{i=0}^{\log L} \frac{2^i}{L} \leq \frac{O(1)}{L} (2) = \frac{O(1)}{L}.$$

□

This completes the proof that our algorithm computes a quasi-uniform cover.

6.3 Derandomization

In this subsection, we note that our algorithm can be derandomized, and thus all the results in this chapter hold deterministically. To our knowledge, derandomization of Varadarajan's technique [Var10] has not been observed before.

The idea is to replace the need for totally independent random choices with b -wise independent random choices for some constant b . Specifically, at the beginning of the algorithm, we generate a b -wise independent sequence of N integers X_1, \dots, X_N that are uniformly distributed in the range $[0, U)$ for a sufficiently large universe size U (for example, $U = \Theta(N)$ would suffice). The subset of columns that are retained during phase i is supposedly a uniform sample of the set of original columns, with a certain sampling probability P_i . To produce this sample, we take the subset of all columns S such that $X_S \in [0, P_i U)$. By a well known construction [Jof74, MR95], a b -wise independent sequence $X_1, \dots, X_N \in [0, U)$ can be generated from b truly random integers in $[0, U)$ for a given prime $U \geq N$. Deterministically, we can try all $O(U^b)$ possible choices for these b integers and thus simulate the randomized algorithm by brute force in polynomial time.

It remains to show that this version of the algorithm using X_1, \dots, X_N still achieves the same expected bound on the weight of the computed set cover. For the analysis, we use the following alternative to the Chernoff bound (e.g., see [SSS95, Theorem 4(III)]): if Z is a sum of b -wise independent 0-1 random variables with $E[Z] = \mu$ for an even b , then

$$\Pr[|Z - \mu| \geq t] \leq \left(\frac{b \cdot \max\{b, \mu\}}{e^{2/3} t^2}\right)^{b/2}.$$

In the proof of Claim 6.2.7, our application has $\mu = k \cdot (1/2 + h(N, k))$ and $t = kh(N, k)$. Thus,

$$\Pr[Z \leq k/2] \leq O\left(\frac{1}{kh(N, k)^2}\right)^{b/2}.$$

Taking $h(N, k) = 1/k^{1/3}$, for example, we can bound the right-hand side by $O(1/k^{b/6})$. Choosing b sufficiently large (as a function of c), we observe that Claim 6.2.7 remains true. Claim 6.2.8 also remains true for our new choice of $h(N, k)$, by suitably replacing square roots with cube roots in the calculations. The final expected weight bound on the set cover follows by linearity of expectation, which does not require independence.

6.4 Geometric Set Systems with Low Shallow Cell Complexity

Our results refine the sampling algorithm by Varadarajan [Var10], and thus there are naturally numerous consequences for geometric covering problems. A standard technique by Clarkson and Shor [CS89] links the shallow cell complexity to union complexity: for a family of objects in a constant dimension d with constant description complexity, if the union complexity is $O(n\phi(n))$, then the shallow cell complexity is bounded by $O((n/k)\phi(n/k) \cdot k^d) \leq O(n\phi(n)k^{d-1})$. We immediately obtain the following corollary of Theorem 6.2.1, which matches and generalizes previous results for the unweighted case [AES10, Var09]:

Corollary 6.4.1. *Let \mathbf{C} be a class of geometric set cover instances where the union of n objects has complexity $O(n\phi(n))$. Then there is a randomized polynomial-time LP-relative $O(\max\{1, \log \phi(n)\})$ -approximation algorithm for the weighted set cover problem for \mathbf{C} .*

This corollary enables us to resolve several of the open questions posed in [Var10] pertaining to weighted geometric set cover for objects with linear or near linear union complexity.

For example, *fat triangles* have been shown to have union complexity $O(n \cdot 2^{\alpha(n)} \log^* n)$ by a recent result of Ezra, Aronov, and Sharir [EAS11] (with de Berg, these authors have apparently improved the bound to $O(n \log^* n)$). Applying Varadarajan's result readily gives a $2^{O(\log^* n)}$ -approximation for the weighted version of \mathbb{R}^2 -**Fat-Triangle-Cover**. Corollary 6.4.1 immediately implies the following strengthening:

Corollary 6.4.2. *There is a randomized poly-time LP-relative $O(\log \log^* n)$ -approximation algorithm for weighted \mathbb{R}^2 -**Fat-Triangle-Cover**.*

Several other examples shall follow. It is well-known that a collection of n disks (or pseudodisks) has union complexity $O(n)$ [KLPS86]. Corollary 6.4.1 thus gives us an $O(1)$ approximation for weighted disk cover, improving the $2^{O(\log^* n)}$ -approximation obtained in [Var10].

The same thing works for axis-aligned octants and unit cubes in \mathbb{R}^3 , since they have linear union complexity [BSTY98]; we thus obtain $O(1)$ approximations for \mathbb{R}^3 -**Unit-Cube-SC** and \mathbb{R}^3 -**3-Sided-Box-SC** as well. Since these set systems are both self-dual (see Proposition 2.3.20), these results extend to \mathbb{R}^3 -**Unit-Cube-HS** and **Min-Hitting-Set**33. Half-spaces in \mathbb{R}^3 have linear union complexity, since the union is the complement of a convex polyhedron. By self-duality again [dBCvKO08], we obtain results for both \mathbb{R}^3 -**Halfspace-SC** and \mathbb{R}^3 -**Halfspace-HS**. Disks in \mathbb{R}^2 can be mapped to half-spaces in \mathbb{R}^3 by the lifting transformation (see Proposition 2.3.17), and so we get an $O(1)$ -approximation for weighted hitting set for disks in the plane. In summary:

Corollary 6.4.3. *There are randomized polynomial-time $O(1)$ -approximation algorithms for the weighted versions of the following covering problems:*

- \mathbb{R}^3 -**Unit-Cube-SC** (equivalently, \mathbb{R}^3 -**Unit-Cube-HS**),
- \mathbb{R}^3 -**Halfspace-SC** (equivalently, \mathbb{R}^3 -**Halfspace-HS**),
- \mathbb{R}^2 -**Disk-SC** (equivalently, \mathbb{R}^2 -**Disk-HS**),
- more generally, \mathbb{R}^2 -**Pseudodisk-SC** and \mathbb{R}^2 -**Pseudodisk-HS**,
- \mathbb{R}^3 -**3-Sided-Box-SC**,
- \mathbb{R}^3 -**3-Sided-Box-HS**.

Recently, Gibson and Pirwani [GP10] have applied Varadarajan’s technique to the weighted *dominating set* problem in the intersection graph of a set of disks in \mathbb{R}^2 . We can map a disk σ with center (a, b) and radius c to a point $p_\sigma = (a, b, c)$, and a disk σ' with center (a', b') and radius c' to a region $S_{\sigma'} = \{(x, y, z) : \sqrt{(x - a')^2 + (y - b')^2} \leq z + c'\}$, so that the two disks intersect if and only if p_σ is covered by $S_{\sigma'}$. The union of the $S_{\sigma'}$ ’s corresponds to a planar additively weighted Voronoi diagram, which is known to have linear complexity [Aur91]. We immediately obtain the following improvement to Gibson and Pirwani’s result:

Corollary 6.4.4. *There is a randomized polynomial-time $O(1)$ -approximation algorithm for weighted \mathbb{R}^2 -**Disk-DS**.*

6.5 Shallow Cell Complexity of Other Set Systems

Here we obtain some results on the shallow cell complexity of non-geometric set systems, such as those of priority covering systems and **Tree-SC**. Our main application is an $O(1)$ -approximation for weighted priority tree cover. We first obtain an exact bound on the SCC of **Tree-SC** matrices:

We first obtain an exact bound on the SCC of **Tree-SC** matrices. In what follows, we are essentially borrowing some ideas from matroid theory without explicitly referring to them. We first need a definition:

Definition 6.5.1. [BL73, Ox111, §6.4] Let $G = (V, E)$ be a connected graph and let $T \subseteq E$ be a tree spanning G . The T -fundamental-cycle incidence matrix of G is the $T \times (E \setminus T)$ matrix A where A_{ef} is 1 if e lies on the fundamental cycle of f in T and 0 otherwise.

Note that tree-fundamental-cycle incidence matrices are precisely the incidence matrices associated with **Tree-SC**.

The following lemma is well-known to matroid theorists [Jae79, Ox111, §14.10]; we prove it here for completeness without reference to matroids:

Lemma 6.5.2. *Let A be a **Tree-SC** matrix. Then the number of distinct rows of A is at most $\max\{3N - 3, 2\}$. In particular, **Tree-SC** has $O(N)$ SCC, since **Tree-SC** is a hereditary class of set systems.*

Proof. If every row of A has at most a single one, then A has at most $N + 1$ distinct rows, and we are done. Suppose A has some row with at least two ones.

We shall assume that A is the T -fundamental-cycle incidence matrix of a graph G for some spanning tree T of G . We first create a new graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and a new tree $\tilde{T} \subseteq \tilde{E}$ spanning \tilde{E} in the following way. While G has a vertex $v \in V$ of degree at most 2, contract one of its incident T edges. In a moment, we will justify why G maintains at least 2 vertices throughout. If we let \tilde{A} be the \tilde{T} -fundamental-cycle incidence matrix of \tilde{G} , observe that we obtained \tilde{A} from A by deleting some zero rows, some rows with exactly one 1, and some duplicate rows. Thus, the number of distinct rows dropped by at most $N + 1$. Note that it also follows that \tilde{A} has at least one row, so $\tilde{T} \neq \emptyset$, which is why G had at least 2 vertices throughout the contraction process.

Since \tilde{G} has minimum degree 3,

$$3|\tilde{V}| \leq 2|\tilde{E}|,$$

so

$$|\tilde{V}| - 1 \leq 2[|\tilde{E}| - (|\tilde{V}| - 1)] - 3 = 2N - 3,$$

which shows \tilde{A} has at most $2N - 3$ rows. Therefore, A has at most $3N - 2$ distinct rows. \square

It turns out that when an SCC bound holds independently of k , we may add priorities at loss of a factor of k in the SCC. More generally, we have the following result:

Lemma 6.5.3. *Let \mathbf{C} be a class of set systems with SCC $f(n, k)$. Then \mathbf{C}^P has SCC at most $kf(n, n)$.*

Proof. Suppose $A \in \mathbf{C}$ and consider what happens when priorities π and \mathbf{s} are added to A , possibly after duplicating some rows of A and applying different demand priorities to each. For simplicity, order the columns of A by decreasing supply priority.

Consider a row x of A . The effect of adding a priority to x is to set some suffix of its corresponding row to zero. Thus, each cell of A induces at most one cell

of each depth when priorities are added—in particular, at most k cells of depth k or fewer (excluding cells containing no ones, which can be ignored). Since A certainly has at most $f(N, N)$ total cells (where N is the number of columns of A), it follows that after adding priorities, A will have at most $kf(N, N)$ cells of depth at most k . The result follows. \square

Lemma 6.5.3 implies the following:

- If \mathbf{C} has $\text{SCC}f(n, k) = g(n)$ for some function g *not* depending on k , then \mathbf{C}^P has $\text{SCC} kg(n)$.
- **Priority-Tree-SC** has $O(nk)$ SCC (via the above point and Lemma 6.5.2).

Combining this with our algorithm, we obtain the following:

Theorem 6.5.4. *There is a randomized polynomial-time $O(1)$ -approximation algorithm for weighted **Priority-Tree-SC**.*

Unfortunately, not all binary TUM matrices have low SCC after adding priorities:

Example 6.5.5. Even transposes of 0, 1 network matrices, which are totally unimodular, can have $\text{SCC} \Omega(n^2)$ after adding priorities. We show this using a set cover problem where the elements are vertical paths in a rooted tree and each set consists of the paths meeting at a specified edge and not exceeding the priority of the edge.

Fix $\ell \geq 1$. Let v_0, v_1, \dots, v_ℓ be a path with ℓ edges, and let w_1, \dots, w_ℓ be leaves, each adjacent to v_0 . For each $1 \leq i \leq \ell$, assign priority i to edge $v_{i-1}v_i$, and assign priority ℓ to edge $w_i v_0$. Root the tree at v_ℓ .

The resulting set cover problem has $n = 2\ell$ sets, and we claim the number of cells of depth 2 is at least $\ell^2 = \frac{n^2}{4}$. Indeed, let $1 \leq i, j \leq \ell$, and consider the path P_{ij} from w_i to v_j having priority j . The only two edges whose corresponding sets contain P_{ij} are $w_i v_0$ and $v_{j-1} v_j$. Thus, each P_{ij} lies in a distinct depth-2 cell.

Chapter 7

Capacitated and Priority Covering Problems

In this chapter, we focus primarily on nongeometric covering problems, most of which arise in applications from combinatorial optimization. A key difference between the results in this chapter and the previous chapters is that we shall obtain results for problems that are not, strictly speaking, special cases of **Min-Set-Cover**. Instead, our primary results shall involve, more general ‘covering’-type problems. Specifically, we will discuss the following three types of problems:

- *Priority covering problems*, defined in Chapter 2, which are special cases of **Min-Set-Cover** obtained from other set systems by the addition of priority supplies and demands.
- *Multi-cover problems*—covering problems in which each element must be covered a specified number of times, which may differ from element to element. These cannot be encoded by **Min-Set-Cover**.
- *Capacitated covering problems*—a generalization of multi-cover problems in which each element is given an integral *demand capacity*, and each set is given an integral *supply capacity*. Each element must be covered by sets whose supply capacities sum to the element’s demand capacity, or greater.

Of course, like **Min-Set-Cover**, all of the above problems come in both weighted and unweighted flavours.

Our key result is an approximation algorithm for capacitated covering problems that uses LP-relative approximation algorithms for priority covering and multi-cover problems as a subroutine. Using the theory we have already developed for priority covering problems in conjunction with some basic results about multi-cover problems, we are able to obtain powerful results for many difficult capacitated covering problems. One of our major results is an $O(1)$ -approximation for the capacitated version of **Tree-SC**. We also list various

algorithmic results for weighted and unweighted capacitated and priority covering problems on lines and trees. Additional details (including integrality gap results) may be found in [CGK10b].

Most of the results in this chapter were originally published in [CGK10a] or the full version [CGK10b]. The final proof that the capacitated version of **Tree-SC** admits a constant approximation requires our quasi-uniform sampling result from Chapter 6, which can also be found in [CGKS12].

7.1 Preliminaries and Statement of Main Result

In a *covering integer program* (CIP), we are given an M by N non-negative constraint matrix A , demands $b \in \mathbb{Z}_+^M$, non-negative costs $c \in \mathbb{Z}_+^N$, and upper bounds $d \in \mathbb{Z}_+^N$. The goal is to solve the following integer linear program (which we denote by $\text{Cov}(A, b, c, d)$):

$$\min\{c^T x : Ax \geq b, 0 \leq x \leq d, x \text{ integer}\}.$$

CIPs generalize the integer programming formulation (SCIP) of **Min-Set-Cover** and are sufficiently powerful to encode capacitated covering and multi-covering (see [Vaz01] for additional applications). Capacitated covering problems can be encoded by forcing all of the non-zero entries of any column of the constraint matrix A to be equal. The resulting CIPs are known as *column-restricted* CIPs (or CCIPs for short). To encode multi-cover problems, we apply a stronger restriction, forcing A to be a binary matrix; the resulting CIPs are known as 0,1-CIPs. Ordinary **Min-Set-Cover** can be encoded by implying yet another strengthening, forcing A to be a binary matrix as well as forcing b and d to both consist entirely of ones. In the *unweighted* versions of all of these problems, the cost vector c is forced to consist entirely of ones.

Weighted CIPs are hard to solve even when they contain just a single row. A capacitated covering problem in which the goal is to cover a single element is precisely equivalent to the knapsack problem; consequently, even CIPs having a single row are **NP**-hard to solve exactly.

General CIPs provide an extremely powerful framework for encoding covering-type problems, and there is a rich and long line of work ([Dob82, Hoc82, RV93, Sri99, Sri06]) on approximation algorithms for them. As a backdrop to what we do in the remainder of this chapter, we shall briefly discuss some of the relevant contributions (though we will not make use of any of these results directly). Assuming no upper bounds on the variables, Srinivasan [Sri99] gave a $O(1 + \log \alpha)$ -approximation for general CIPs, where α , called the *dilation* of the instance, denotes the maximum number of non-zero entries in any column of the constraint matrix. Later on, Kolliopoulos and Young [KY05] obtained the same approximation factor, respecting the upper bounds. However, these algorithms do not give any better results when special structure of the constraint matrix is known. Even for the special case of CCIPs, nothing better is known unless one aims for *bicriteria* results where solutions violate the upper bound constraints $x \leq d$. On the hardness side, Trevisan [Tre01] showed that it is

NP-hard to obtain a $(\log \alpha - O(\log \log \alpha))$ -approximation, even for the special case of 0,1-CIPs.

As with the set cover problems discussed in previous chapters, our method will be to exploit additional underlying structure to better solve CCIPs and 0,1-CIPs. As a warm-up, we observe that CIPs remain solvable in polynomial time whenever the constraint matrix A is totally unimodular, since the canonical LP relaxation of the CIP is then integral (e.g., see [Sch03]). Consequently, the multi-cover versions of **Network-SC** and **Network-HS** remain polynomial-time solvable and have an integrality gap of 1 (by Theorem 2.1.18).

While a number of general techniques have been developed for obtaining improved approximation algorithms for structured covering problems like the ones discussed in previous chapters, very little is known for structured CIP instances that are *not* 0,1. Our study of CCIPs in this chapter is our attempt to mitigate this problem. The main focus of the line of research discussed herein is to understand how the structure of the underlying 0,1-CIP can be used to derive improved approximation algorithms for CCIPs. Our primary result is the following:

Theorem 7.1.1. *Let \mathbf{C} be a class of set systems. Suppose that the following assumptions hold:*

- A1 *The weighted 0,1-CIP on \mathbf{C} has integrality gap $\gamma \geq 1$. In other words, among all of the multi-cover problems obtained by taking an (SCIP) instance on some matrix in \mathbf{C} and adding non-negative vectors of demands and weights, none have integrality gap greater than γ .*
- A2 *The weighted PCIP problem on \mathbf{C} (that is, the regular weighted set cover problem on the class \mathbf{C}^P of set systems) has integrality gap $\omega \geq 1$.*

Then the weighted CCIP on \mathbf{C} admits a polynomial time LP-relative $(24\gamma + 8\omega)$ -approximation algorithm for column-restricted CIPs. In other words, there is an $O(\gamma + \omega)$ -approximation algorithm for capacitated covering problems obtained by taking element-set incidence matrices from \mathbf{C} and adding arbitrary supply capacities and costs to sets and arbitrary demand capacities to elements.

The main idea implicit in Theorem 7.1.1 is that the approximability of a class of CCIP instances (defined on some family of set systems) is related to the integrality gap of the underlying PCIP and 0,1-CIP problems (defined on the same family of set systems). Accordingly, capacitated covering problems can be approximated well whenever their underlying priority covering problems and multi-cover problems admit good LP-relative approximation algorithms.

Additionally, we highlight the following key aspects of this result:

- The variables γ and ω need not be fixed integers; indeed, they may vary according to any instance-related parameter, allowing non-constant approximation algorithms to be obtained via our method.
- We do not explicitly require an *algorithm* to solve the underlying PCIP or 0,1-CIP problems in order to obtain our approximation for the CCIPs. Any proof of a small integrality gap is sufficient.

7. CAPACITATED AND PRIORITY COVERING PROBLEMS

- Conversely, since we do require an integrality gap proof for the underlying PCIP and 0,1-CIP problems, approximation algorithms for PCIPs or 0,1-CIPs that are *not* LP-relative are insufficient for applying the theorem.
- Theorem 7.1.1 also holds for unweighted CCIPs. Specifically, to obtain a $(24\gamma + 8\omega)$ -approximation for *unweighted* capacitated covering problems on a class \mathbf{C} of set systems, it suffices to prove integrality gap bounds of ω and γ on the *unweighted* priority and multi-cover problems on \mathbf{C} .
- As part of our analysis, we will prove that an integrality gap of ω need only be obtained for PCIPs in which the number of distinct priorities is $O(\log s_{\max})$ where s_{\max} is the largest integer supply priority in the original CCIP instances we wish to approximate. This has implications for some applications.

Theorem 7.1.1 is proven using LP strengthening and rounding. We add a set of valid constraints called the *knapsack cover constraints* to the canonical LP relaxation of a CCIP, which can otherwise have an unbounded integrality gap. We then give a method of rounding a fractional solution to the resulting LP by relating it to LPs obtained from the related PCIP and 0,1-CIP problems.

Our method is related to a prior result of Kolliopoulos [Kol03]. The author studies CCIPs that satisfy a rather strong assumption, called the *no bottleneck assumption*, which implies that the supply of any column is smaller than the demand of any row. Kolliopoulos [Kol03] shows that if one is allowed to violate the upper bounds by a multiplicative constant, then the integrality gap of the CCIP is within a constant factor of that of the original 0,1-CIP.¹ As the author notes, such a violation is necessary; otherwise the CCIP has unbounded integrality gap. If one is not allowed to violate upper bounds, nothing better than the result of [KY05] is known for the special case of CCIPs.

Our work on CCIPs parallels a large body of work on column-restricted *packing* integer programs (CPIPs), the ‘dual’ problem to covering. Assuming the *no-bottleneck assumption*, Kolliopoulos and Stein [KS04] show that CPIPs can be approximated asymptotically as well as the corresponding 0,1-PIPs. Chekuri et al. [CMS07] subsequently improve the constants in the result from [KS04]. These results imply constant factor approximations for the column-restricted tree *packing* problem under the no-bottleneck assumption. Without the no-bottleneck assumption, however, only a polylogarithmic approximation is known for the problem [CEK09].

The only work on priority versions of covering problems that we are aware of is due to Charikar, Naor and Schieber [CNS04] who studied the priority Steiner tree and forest problems in the context of Quality of Service (QoS) management in a network multicasting application. Charikar et al. present an $O(\log n)$ -approximation algorithm for the problem, and Chuzhoy et al. [CGNS08] later show that no efficient $o(\log \log n)$ approximation algorithm can exist unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$ (n is the number of vertices).

¹Such a result is implicit in the paper; the author only states a $O(\log \alpha)$ integrality gap.

To the best of our knowledge, the column-restricted or priority versions of the line and tree cover problem have not been studied. The best known approximation algorithm known for both is the $O(\log n)$ factor implied by the results of [KY05] stated above. However, upon completion of our work, Nitish Korula [Kor09] pointed out to us that a 4-approximation for column-restricted line cover is implicit in a result of Bar-Noy et al. [BNBYF⁺01]. We remark that their algorithm is not LP-relative, although our general result on CCIPs is. Moreover, their results cannot be extended to trees.

7.2 Proof of Main Result

In this section, we shall prove Theorem 7.1.1 via LP strengthening and rounding. Throughout this section, we shall fix a class \mathbf{C} of set systems and assume that assumptions A1 and A2 from above hold. We begin with an overview of the method.

The key ingredient in the proof is the addition of *knapsack cover constraints* to the standard LP relaxation of the CCIP. Knapsack cover constraints were first used to strengthen LP relaxations in [Bal75, HJP75, Wol75]; Carr et al. [CFLP00] were the first to use them in the design approximation algorithms. The paper of Kolliopoulos and Young [KY05] also uses these to obtain their result on general CIPs.

The main technique used for designing algorithms for column-restricted problems is *grouping-and-scaling* developed by Kolliopoulos and Stein [KS01, KS04] for packing problems, and later used by Kolliopoulos [Kol03] in the covering context. In this technique, the *columns* of the matrix are divided into groups of ‘similar’ supply values; in a single group, the supply values are then scaled to be the same; for a single group, the integrality gap of the original 0,1-CIP is invoked to get an integral solution for that group; the final solution is a ‘union’ of the solutions over all groups.

There are two issues in applying the technique to the new strengthened LP relaxation of our problem. First, although the original constraint matrix is column-restricted, the new constraint matrix with the knapsack cover constraints is not. Secondly, unless additional assumptions are made, the current grouping-and-scaling analysis does not give a handle on the degree of violation of the upper bound constraints. This is the reason why Kolliopoulos [Kol03] needs the strong no-bottleneck assumption.

We get around the first difficulty by grouping the *rows* as well, into those that get most of their coverage from columns not affected by the knapsack constraints, and the remainder. On the first group of rows, we apply a subtle modification to the vanilla grouping-and-scaling analysis and obtain an $O(\gamma)$ -approximate feasible solution satisfying these rows; we then show that one can treat the remainder of the rows as a PCIP and get an $O(\omega)$ -approximate feasible solution satisfying them, using assumption A2. Combining the two gives the $O(\gamma + \omega)$ factor. The full details are given in the Subsections that follow.

7.2.1 Strengthening the Canonical LP Relaxation

In the following, we use \mathcal{C} for the set of columns and \mathcal{R} for the set of rows of A . For a vector $s \in \mathbb{R}^N$, we denote by $A[s]$ a column-restricted constraint matrix whose j^{th} column contains only elements in $\{0, s_j\}$.

Let $F \subset \mathcal{C}$ be a subset of the columns in the CCIP $\text{Cov}(A[s], b, c, d)$. For all rows $i \in \mathcal{R}$, define $b_i^F = \max\{0, b_i - \sum_{j \in F} A[s]_{ij} d_j\}$ to be the residual demand of row i w.r.t. F . Define matrix $A^F[s]$ by letting

$$A^F[s]_{ij} = \begin{cases} \min\{A[s]_{ij}, b_i^F\} & : j \in \mathcal{C} \setminus F \\ 0 & : j \in F, \end{cases} \quad (7.2.1)$$

for all $i \in \mathcal{C}$ and for all $j \in \mathcal{R}$. The following *Knapsack-Cover* (KC) inequality

$$\sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j \geq b_i^F$$

is valid for the set of all integer solutions x for $\text{Cov}(A[s], b, c, d)$. Adding the set of all KC inequalities yields the following stronger LP formulation CIP. We note that the LP is not column-restricted, in that, different values appear on the same column of the new constraint matrix.

$$\begin{aligned} \text{opt}_P &:= \min \sum_{j \in \mathcal{C}} c_j x_j & (P) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j \geq b_i^F & \forall F \subseteq \mathcal{C}, \forall i \in \mathcal{R} \\ & 0 \leq x_j \leq d_j & \forall j \in \mathcal{C} \end{aligned} \quad (7.2.2)$$

It is not known whether (P) can be solved in polynomial time. For $\alpha \in (0, 1)$, call a vector x^* α -relaxed if its cost is at most opt_P , and if it satisfies (7.2.2) for $F = \{j \in \mathcal{C} : x_j^* \geq \alpha d_j\}$. An α -relaxed solution to (P) can be computed efficiently for any α . To see this note that one can check whether a candidate solution satisfies (7.2.2) for a set F ; we are done if it does, and otherwise we have found an inequality of (P) that is violated, and we can make progress via the ellipsoid method. Details can be found in [CFLP00] and [KY05].

We fix an $\alpha \in (0, 1)$, specifying its precise value later. Compute an α -relaxed solution, x^* , for (P), and let $F = \{j \in \mathcal{C} : x_j^* \geq \alpha d_j\}$. Define \bar{x} as, $\bar{x}_j = x_j^*$ if $j \in \mathcal{C} \setminus F$, and $\bar{x}_j = 0$, otherwise. Since x^* is an α -relaxed solution, we get that \bar{x} is a feasible fractional solution to the *residual* CIP, $\text{Cov}(A^F[s], b^F, c, \alpha d)$. In the next subsection, our goal will be to obtain an *integral* feasible solution to the covering problem $\text{Cov}(A^F[s], b^F, c, d)$ using \bar{x} . The next lemma shows how this implies an approximation to our original CIP.

Lemma 7.2.1. *If $\text{Cov}(A^F[s], b^F, c, d)$ admits an integral feasible solution, x^{int} with $c^T x^{\text{int}} \leq \beta \cdot c^T \bar{x}$, then there exists a $\max\{1/\alpha, \beta\}$ -factor approximation to $\text{Cov}(A[s], b, c, d)$.*

Proof. Define

$$z_j = \begin{cases} d_j & : j \in F \\ x_j^{\text{int}} & : j \in \mathcal{C} \setminus F, \end{cases} \quad (7.2.3)$$

Observe that $z \leq d$. z is a feasible integral solution to $\text{Cov}(A[s], b, c, d)$ since for any $i \in \mathcal{R}$,

$$\begin{aligned} \sum_{j \in \mathcal{C}} A[s]_{ij} z_j &= \sum_{j \in F} A[s]_{ij} d_j + \sum_{j \in \mathcal{C} \setminus F} A[s]_{ij} x_j^{\text{int}} \\ &\geq (b_i - b_i^F) + \sum_{j \in \mathcal{C} \setminus F} A^F[s]_{ij} x_j^{\text{int}} \\ &\geq b_i \end{aligned}$$

where the first inequality follows from the definition of b_i^F and since $A[s]_{ij} \geq A^F[s]_{ij}$, the second inequality follows because x^{int} is a feasible solution to the instance $\text{Cov}(A^F[s], b^F, c, d)$.

Furthermore,

$$c^T z = \sum_{j \in F} c_j d_j + \sum_{j \in \mathcal{C} \setminus F} c_j x_j^{\text{int}} \leq \frac{1}{\alpha} \sum_{j \in F} c_j x_j^* + \beta \sum_{j \in \mathcal{C} \setminus F} c_j x_j^* \leq \max\{\frac{1}{\alpha}, \beta\} \text{opt}_P$$

where the first inequality follows from the definition of F and the second from the assumption in the theorem statement. \square

7.2.2 Solving the Residual Problem

In this section we use a feasible fractional solution \bar{x} of $\text{Cov}(A^F[s], b^F, c, \alpha d)$, to obtain an *integral* feasible solution x^{int} to the problem $\text{Cov}(A^F[s], b^F, c, d)$, with $c^T x^{\text{int}} \leq \beta c^T \bar{x}$ for $\beta = 24\gamma + 8\omega$. Fix $\alpha = 1/24$.

Converting to Powers of 2. For ease of exposition, we first modify the input to the residual problem $\text{Cov}(A^F[s], b^F, c, d)$ so that all entries of b^F are powers of 2. For every $i \in \mathcal{R}$, let \bar{b}_i denote the smallest power of 2 larger than b_i^F . For every column $j \in \mathcal{C}$, let \bar{s}_j denote the largest power of 2 smaller than s_j .

Lemma 7.2.2. $y = 4\bar{x}$ is feasible for $\text{Cov}(A^F[\bar{s}], \bar{b}, c, 4\alpha d)$.

Proof. Focus on row $i \in \mathcal{R}$. We have

$$\sum_{j \in \mathcal{C}} A^F[\bar{s}]_{ij} y_j \geq 2 \cdot \sum_{j \in \mathcal{C}} A^F[s]_{ij} \bar{x}_j \geq 2b_i^F \geq \bar{b}_i,$$

where the first inequality uses the fact that $s_j \leq 2\bar{s}_j$ for all $j \in \mathcal{C}$, the second inequality uses the fact that \bar{x} is feasible for $\text{Cov}(A^F[s], b^F, c, \alpha d)$, and the third follows from the definition of \bar{b}_i . \square

Partitioning the rows. We call \bar{b}_i the residual demand of row i . For a row i , a column $j \in \mathcal{C}$ is *i-large* if the supply of j is at least the residual demand of row i ; it is *i-small* otherwise. Formally,

$$\begin{aligned} \mathcal{L}_i &= \{j \in \mathcal{C} : A_{ij} = 1, \bar{s}_j \geq \bar{b}_i\} \text{ is the set of } i\text{-large columns} \\ \mathcal{S}_i &= \{j \in \mathcal{C} : A_{ij} = 1, \bar{s}_j < \bar{b}_i\} \text{ is the set of } i\text{-small columns} \end{aligned}$$

Recall the definition from (7.2.1), $A^F[\bar{s}]_{ij} = \min(A[\bar{s}]_{ij}, b_i^F)$. Therefore, the entry $A^F[\bar{s}]_{ij} = A_{ij}b_i^F$ for all $j \in \mathcal{L}_i$ since $\bar{s}_j \geq \bar{b}_i \geq b_i^F$; and $A^F[\bar{s}]_{ij} = A_{ij}\bar{s}_j$ for all $j \in \mathcal{S}_i$, since being powers of 2, $\bar{s}_j < \bar{b}_i$ implies, $\bar{s}_j \leq \bar{b}_i/2 \leq b_i^F$.

We now partition the rows into large and small depending on which columns most of their coverage comes from. Formally, call a row $i \in \mathcal{R}$ *large* if

$$\sum_{j \in \mathcal{S}_i} A^F[\bar{s}]_{ij} y_j \leq \sum_{j \in \mathcal{L}_i} A^F[\bar{s}]_{ij} y_j,$$

and small otherwise. Note that Lemma 7.2.2 together with the fact that each column in row i 's support is either small or large implies,

$$\text{For a large row } i, \quad \sum_{j \in \mathcal{L}_i} A^F[\bar{s}]_{ij} y_j \geq \bar{b}_i/2.$$

$$\text{For a small row } i, \quad \sum_{j \in \mathcal{S}_i} A^F[\bar{s}]_{ij} y_j \geq \bar{b}_i/2.$$

Let \mathcal{R}_L and \mathcal{R}_S be the set of large and small rows.

In the following, we address small and large rows separately. We compute a pair of integral solutions $x^{\text{int},S}$ and $x^{\text{int},\mathcal{L}}$ that are feasible for the small and large rows, respectively. We then obtain x^{int} by letting

$$x_j^{\text{int}} = \max\{x_j^{\text{int},S}, x_j^{\text{int},\mathcal{L}}\}, \quad (7.2.4)$$

for all $j \in \mathcal{C}$.

7.2.3 Dealing with Small Rows via Multi-Cover

For these rows we use the grouping-and-scaling technique similar to that of [CMS07, Kol03, KS01, KS04]. However, as mentioned in the introduction, we use a modified analysis that bypasses the no-bottleneck assumptions made by earlier works.

Lemma 7.2.3. *We can find an integral solution $x^{\text{int},S}$ such that*

- a) $x_j^{\text{int},S} \leq d_j$ for all j ,
- b) $\sum_{j \in \mathcal{C}} c_j x_j^{\text{int},S} \leq 24\gamma \sum_{j \in \mathcal{C}} c_j \bar{x}_j$, and
- c) for every small row $i \in \mathcal{R}_S$, $\sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int},S} \geq b_i^F$.

Proof. The complete proof is slightly technical and hence we start with a sketch.

Since the rows are small, for any row i , we can zero out the entries that are larger than \bar{b}_i , and still $2y$ will be a feasible solution. Note that, now in each

row, the entries are $< \bar{b}_i$, and thus are at most $\bar{b}_i/2$ (everything being powers of 2). We stress that it could be that \bar{b}_i of some row is less than the entry in some other row, that is, we do not have the no-bottleneck assumption. However, when a particular row i is fixed, \bar{b}_i is at least any entry of the matrix in the i th row. Our modified analysis of grouping and scaling then makes the proof go through.

We *group* the columns into classes that have s_j as the same power of 2, and for each row i we let $\bar{b}_i^{(t)}$ be the contribution of the class t columns towards the demand of row i . The columns of class t , the small rows, and the demands $\bar{b}_i^{(t)}$ form a CIP where all non-zero entries of the matrix are the same power of 2. We scale both the constraint matrix and $\bar{b}_i^{(t)}$ down by that power of 2 to get a 0,1-CIP, and using assumption A1, we get an integral solution to this 0,1-CIP. Our final integral solution is obtained by concatenating all these integral solutions over all classes.

Till now the algorithm is the standard grouping-and-scaling algorithm. The difference lies in our analysis in proving that this integral solution is feasible for the original CCIP. Originally the no-bottleneck assumption was used to prove this. However, we show since the column values in different classes are geometrically decreasing, the weaker assumption of \bar{b}_i being at least any entry in the i th row is enough to make the analysis go through. We now get into the full proof.

Step 1: Grouping the columns

Let \bar{s}_{min} and \bar{s}_{max} be the smallest and largest supply among the columns in $\mathcal{C} \setminus F$. Since all \bar{s}_j are powers of 2, we introduce the shorthand, $\bar{s}^{(t)}$ for the supply $\bar{s}_{max}/2^t$. We say that a column j is in *class* $t \geq 0$, if $\bar{s}_j = \bar{s}^{(t)}$, and we let

$$\mathcal{C}^{(t)} := \{j \in \mathcal{C} \setminus F : \bar{s}_j = \bar{s}^{(t)}\}$$

be the set of class t supplies.

Step 2: Disregarding i -large columns of a small row i

Fix a small row $i \in \mathcal{R}_S$. We now identify the columns j that are i -small. To do so, define $t_i := \log(\bar{s}_{max}/\bar{b}_i) + 1$. Observe that any column j in class $\mathcal{C}^{(t)}$ for $t \geq t_i$ are i -small. This is because $\bar{s}_j = s_{max}/2^t \leq s_{max}/2^{t_i} = \bar{b}_i/2 < \bar{b}_i$. Define

$$\bar{b}_i^{(t)} = \begin{cases} 2 \sum_{j \in \mathcal{C}^{(t)}} A^F[\bar{s}]_{ij} y_j & : t \geq t_i \\ 0 & : \text{otherwise} \end{cases}$$

as the contribution of the class t , i -small columns to the demand of row i , multiplied by 2. Note that by definition of small rows, these columns contribute to more than 1/2 of the demand, and so

$$\sum_{t \geq t_i} \bar{b}_i^{(t)} \geq \bar{b}_i. \tag{7.2.5}$$

7. CAPACITATED AND PRIORITY COVERING PROBLEMS

Henceforth, we will consider only the contributions of the small i -columns of a small row i .

Step 3: Scaling and getting the integral solution

Fix a class t of columns and scale down by $\bar{s}^{(t)}$ to get a $\{0, 1\}$ -constraint matrix (recall entries of the columns in a class t are all $\bar{s}^{(t)}$). This will enable us to apply assumption A1 and get an integral solution corresponding to these columns. The final integral solution will be the concatenation of the integral solutions over the various classes.

The constants in the next claim are carefully chosen for the calculations to work out later.

Claim 7.2.4. For any $t \geq 0$ and for all $i \in \mathcal{R}_S$, $6 \cdot \sum_{j \in \mathcal{C}^{(t)}} A_{ij} y_j \geq \lfloor 3\bar{b}_i^{(t)} / \bar{s}^{(t)} \rfloor$.

Proof. The claim is trivially true for rows i with $t_i > t$ as $\bar{b}_i^{(t)} = 0$ in this case. Consider a row i with $t_i \leq t$. Since any column $j \in \mathcal{C}^{(t)}$ is i -small, we get $A^F[\bar{s}]_{ij} = A_{ij} \bar{s}_j = A_{ij} \bar{s}^{(t)}$. Using the definition of \bar{b}_i , we obtain

$$6 \cdot \sum_{j \in \mathcal{C}^{(t)}} A_{ij} \bar{s}^{(t)} y_j = 3\bar{b}_i^{(t)}.$$

Dividing both sides by $\bar{s}^{(t)}$ and taking the floor on the right-hand side yields the claim. \square

Since $\alpha = 1/24$ and \bar{x} is a feasible solution to $\text{Cov}(A^F[s], b^F, c, d/24)$, we get that $6y_j = 24 \cdot \bar{x}_j \leq d_j$ for all $j \in \mathcal{C} \setminus F$. Thus, the above claim shows that $6y$ is a feasible fractional solution for $\text{Cov}(A^{(t)}, \lfloor 3\bar{b}_i^{(t)} / \bar{s}^{(t)} \rfloor, c^{(t)}, d^{(t)})$, where $A^{(t)}$ is the submatrix of A defined by the columns in $\mathcal{C}^{(t)}$, and $c^{(t)}$ and $d^{(t)}$ are the sub-vectors of c and d , respectively, that are induced by $\mathcal{C}^{(t)}$. Using assumption A1, we therefore conclude that there is an integral vector $x^{\text{int}, S, t}$ such that

$$x_j^{\text{int}, S, t} \leq d_j \quad \text{for all } j \in \mathcal{C}^{(t)}, \text{ and} \quad (7.2.6)$$

$$\sum_{j \in \mathcal{C}^{(t)}} A_{ij}^{(t)} x_j^{\text{int}, S, t} \geq \left\lfloor \frac{3\bar{b}_i^{(t)}}{\bar{s}^{(t)}} \right\rfloor \quad \text{for all } i \in \mathcal{R}_S, \text{ and} \quad (7.2.7)$$

$$\sum_{j \in \mathcal{C}^{(t)}} c_j x_j^{\text{int}, S, t} \leq 6\gamma \cdot \sum_{j \in \mathcal{C}^{(t)}} c_j y_j \quad (7.2.8)$$

We obtain integral solution $x^{\text{int}, S}$ by letting $x_j^{\text{int}, S} = x_j^{\text{int}, S, t}$ if $j \in \mathcal{C}^{(t)}$. Thus $x_j^{\text{int}, S} \leq d_j$ for all $j \in \mathcal{C}$, and we get,

$$\sum_{j \in \mathcal{C}} c_j x_j^{\text{int}, S} = \sum_{t \geq 0} \sum_{j \in \mathcal{C}^{(t)}} c_j x_j^{\text{int}, S, t} \leq 6\gamma \cdot \sum_{t \geq 0} \sum_{j \in \mathcal{C}^{(t)}} c_j y_j = 24\gamma \cdot \sum_{j \in \mathcal{C}} c_j \bar{x}_j. \quad (7.2.9)$$

Thus we have established parts (a) and (b) of the lemma. It remains to show that $x^{\text{int}, S}$ is feasible for the set of small rows.

Step 4: Putting them all together: scaling back

Once again, fix a small row $i \in \mathcal{R}_S$. The following inequality takes only contribution of the i -small columns. We later show this suffices.

$$\begin{aligned} \sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int}, \mathcal{S}} &\geq \sum_{j \in \mathcal{C}: j \text{ is } i\text{-small}} A_{ij} s_j x_j^{\text{int}, \mathcal{S}} \\ &= \sum_{t \geq t_i} \sum_{j \in \mathcal{C}^{(t)}} A_{ij}^{(t)} s_j x_j^{\text{int}, \mathcal{S}} \geq \sum_{t \geq t_i} \sum_{j \in \mathcal{C}^{(t)}} A_{ij}^{(t)} \bar{s}^{(t)} x_j^{\text{int}, \mathcal{S}, t} \end{aligned} \quad (7.2.10)$$

The first inequality follows since $A^F[s]_{ij} = A_{ij} s_j$ for i -small columns, the equality follows from the definition of t_i , and the final inequality uses the fact that $s_j \geq \bar{s}^{(t)}$ for $j \in \mathcal{C}^{(t)}$. The following claim along with (7.2.10) proves feasibility of row i . This is the part where our analysis slightly differs from the standard grouping-and-scaling analysis.

Claim 7.2.5. *For any small row $i \in \mathcal{R}_S$,*

$$\sum_{t \geq t_i} \sum_{j \in \mathcal{C}^{(t)}} A_{ij}^{(t)} \bar{s}^{(t)} x_j^{\text{int}, \mathcal{S}, t} \geq b_i^F.$$

Proof. In this proof, the choice of the constant 3 on the right-hand side of the inequality in Claim 7.2.4 will become clear. Let $S_i = \{t \geq t_i : 3\bar{b}_i^{(t)} < \bar{s}^{(t)}\}$ be the set of i -small classes t whose fractional supply $\bar{b}_i^{(t)}$ is small compared to its integral supply $\bar{s}^{(t)}$. We now show that for any small row i , the columns in the classes not in S_i suffice to satisfy its demand. Note that

$$\sum_{t \notin S_i, t \geq t_i} \bar{b}_i^{(t)} = \sum_{t \geq t_i} \bar{b}_i^{(t)} - \sum_{t \in S_i} \bar{b}_i^{(t)} \geq \sum_{t \geq t_i} \bar{b}_i^{(t)} - \frac{1}{3} \sum_{t \in S_i} \bar{s}^{(t)} \quad (7.2.11)$$

which follows from the definition of S_i . Furthermore, from (7.2.5) we know that for a small row, $\sum_{t \geq t_i} \bar{b}_i^{(t)} \geq \bar{b}_i$. Also, since $\bar{s}^{(t)}$ form a geometric series, we get that $\sum_{t \in S_i} \bar{s}^{(t)} \leq \sum_{t \geq t_i} \bar{s}^{(t)} \leq 2\bar{s}^{(t_i)}$. Putting this in (7.2.11) we get

$$\sum_{t \notin S_i, t \geq t_i} \bar{b}_i^{(t)} \geq \bar{b}_i - \frac{1}{3} \sum_{t \geq t_i} \bar{s}^{(t)} \geq \bar{b}_i - \frac{2}{3} \bar{s}^{(t_i)} = \frac{2}{3} \bar{b}_i, \quad (7.2.12)$$

where the final equality follows from the definition of t_i , which implies that $\bar{s}^{(t_i)} = \bar{b}_i/2$.

Moreover, for $t \notin S_i$, we know that $\lfloor 3\bar{b}_i^t / \bar{s}^{(t)} \rfloor \geq \frac{3}{2} \bar{b}_i^t / \bar{s}^{(t)}$ since $\lfloor a \rfloor \geq a/2$ if $a > 1$. Therefore, using inequality (7.2.7) in (7.2.10), we get

$$\begin{aligned} \sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int}, \mathcal{S}} &\geq \sum_{t \geq t_i} \sum_{j \in \mathcal{C}^{(t)}} A_{ij}^{(t)} \bar{s}^{(t)} x_j^{\text{int}, \mathcal{S}, t} \geq \sum_{t \notin S_i, t \geq t_i} \bar{s}^{(t)} \left\lfloor \frac{3\bar{b}_i^{(t)}}{\bar{s}^{(t)}} \right\rfloor \\ &\geq \frac{3}{2} \sum_{t \notin S_i, t \geq t_i} \bar{b}_i^{(t)} \\ &\geq \bar{b}_i \geq b_i^F, \end{aligned}$$

where the second-last inequality uses (7.2.12), and the last uses the definition of \bar{b}_i . This completes the proof of the lemma. \square

\square

7.2.4 Dealing with Large Rows via Priority Covering

The large rows can be showed to be a PCIP problem and thus assumption A2 can be invoked to get an analogous lemma to Lemma 7.2.3.

Lemma 7.2.6. *We can find an integral solution $x^{\text{int},\mathcal{L}}$ such that*

- a) $x_j^{\text{int},\mathcal{L}} \leq 1$ for all j ,
- b) $\sum_{j \in \mathcal{C}} c_j x_j^{\text{int},\mathcal{S}} \leq 8\omega \sum_{j \in \mathcal{C}} c_j \bar{x}_j$, and
- c) for every large row $i \in \mathcal{R}_L$, $\sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int},\mathcal{S}} \geq b_i^F$.

Proof. Let $i \in \mathcal{R}_L$ be a large row, and recall that \mathcal{L}_i is the set of i -large columns in \mathcal{C} . We have

$$\sum_{j \in \mathcal{L}_i} A^F[s]_{ij} y_j = \sum_{j \in \mathcal{L}_i} A_{ij} \bar{b}_i y_j \geq \bar{b}_i / 2,$$

and hence

$$2 \sum_{j \in \mathcal{L}_i} A_{ij} y_j \geq 1. \quad (7.2.13)$$

Let $A^{\mathcal{R}}$ be the minor of A induced by the large rows. Consider the priority cover problem $\text{Cov}(A^{\mathcal{R}}[\bar{s}, \bar{b}], \mathbf{1}, c)$. From the definition of \mathcal{L}_i , it follows $2y$ is a feasible fractional solution to the priority cover problem.

Using assumption A2, we conclude that there is an integral solution $x^{\text{int},\mathcal{L}}$ such that $\sum_{j \in \mathcal{C}} c_j x_j^{\text{int},\mathcal{L}} \leq 2\omega \sum_{j \in \mathcal{C}} c_j y_j = 8\omega \sum_{j \in \mathcal{C}} c_j \bar{x}_j$, and $\sum_{j \in \mathcal{C}} A_{ij}^{\mathcal{R}} x_j^{\text{int},\mathcal{L}} \geq 1$, for all large rows $i \in \mathcal{R}_L$.

Fix a large row i . Since $A^F[s]_{ij} = b_i^F$ for all i -large columns \mathcal{L}_i , we get

$$\sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int},\mathcal{L}} \geq \sum_{j \in \mathcal{L}_i} A_{ij} b_i^F x_j^{\text{int},\mathcal{L}} = b_i^F \sum_{j \in \mathcal{C}} A_{ij}^{\mathcal{R}} x_j^{\text{int},\mathcal{L}} \geq b_i^F$$

This completes the proof of the lemma. \square

7.2.5 Putting Everything Together

We now complete the proof of Theorem 7.1.1.

Let $x^{\text{int},\mathcal{S}}$ and $x^{\text{int},\mathcal{L}}$ be as satisfying the conditions of Lemma 7.2.3 and 7.2.6, respectively. Define x^{int} as $x_j^{\text{int}} = \max\{x_j^{\text{int},\mathcal{S}}, x_j^{\text{int},\mathcal{L}}\}$. We have

- a) $x_j^{\text{int}} \leq d_j$ since both $x_j^{\text{int},\mathcal{S}} \leq d_j$ and $x_j^{\text{int},\mathcal{L}} \leq 1 \leq d_j$.
- b) For any row i , $\sum_{j \in \mathcal{C}} A^F[s]_{ij} x_j^{\text{int}} \geq b_i^F$ since the inequality is true with x^{int} replaced by $x^{\text{int},\mathcal{S}}$ for small rows, and x^{int} by $x^{\text{int},\mathcal{L}}$ for large rows.

$$c) \sum_{j \in \mathcal{C}} c_j x_j^{\text{int}} \leq \sum_{j \in \mathcal{C}} c_j x_j^{\text{int}, \mathcal{S}} + \sum_{j \in \mathcal{C}} c_j x_j^{\text{int}, \mathcal{L}} \leq (24\gamma + 8\omega) \sum_{j \in \mathcal{C}} c_j \bar{x}_j.$$

Thus, x^{int} is a feasible integral solution to $\text{Cov}(A^F[s], b^F, c, d)$ with cost bounded as $\sum_{j \in \mathcal{C}} c_j x_j^{\text{int}} \leq (24\gamma + 8\omega) \sum_{j \in \mathcal{C}} c_j \bar{x}_j$. Noting that $\alpha = 1/24$, the proof of the theorem follows from Lemma 7.2.1. \square

7.3 Applications

Bansal et al. [BKS11] very recently showed that if the integrality gap bound γ of the underlying multicover family is *hereditary* in the sense that it also holds for row-induced sub-systems, then the integrality gap of the LP relaxations of the corresponding priority instances is $O(\alpha \log^2 k)$, where k is the number of distinct priorities. In addition, the authors show the hereditary multicover gap is γ whenever a given instance has *hereditary discrepancy* at most α .

One of the main specific questions left open when [CGK10a] was originally published concerns the case where the set system matrix A is a network matrix. Does the addition of capacities make the problem harder in this case? The work in [BKS11] implies an $O((\log \log s_{\max})^2)$ approximation for the capacitated set-cover problem in this case, where s_{\max} is the largest supply. We improve over [BKS11] and settle the open question in [CGK10a].

Theorem 7.3.1. *There is a constant-factor approximation for the weighted capacitated covering problem whenever the underlying set system matrix A is a network matrix.*

Proof. The corresponding multicover problem involving network matrices is totally unimodular and thus admits an integrality gap of 1. Consequently, by Theorem 7.1.1, it suffices to bound the integrality gap of the related priority covering problem. In other words, we must establish a constant factor upper bound on the integrality gap of **Priority-Network-SC**. Fortunately, Theorem 6.5.4 provides an LP-relative $O(1)$ -approximation for it, which implicitly yields a proof that the integrality gap is bounded by a constant. The result follows. \square

Unfortunately, the same argument does not work for all totally unimodular matrices. As shown by Example 6.5.5, there may exist TUM covering problems whose priority versions do not have low SCC and consequently do not admit a good approximation using the methods of Chapter 6.

For non-TUM problems, it is not clear that anything can be done without an improved way to bound the integrality gap of the multi-cover instance. Fortunately, there exist some techniques for dealing with multi-cover problems, and they can be applied to set systems of low SCC:

Theorem 7.3.2. *There is an $O(\log \log N)$ -approximation for the capacitated covering problem whenever the underlying set system matrix has SCC $O(n)$.*

Proof. As in the proof of Theorem 7.3.1, we can use the methods of Chapter 6 to obtain a constant approximation for the priority covering version, so it suffices to obtain an $O(\log \log N)$ -approximation for the multi-cover problem.

Using a trick from [BP10, §5], it is possible to obtain a polynomial-time LP-relative $O(\log \log M)$ -approximation for the multicover version of any standard set cover problem admitting a constant approximation (where M is the number of rows in the set system matrix). However, for problems with linear SCC we have $M \in O(N)$, and thus a $O(\log \log M)$ -approximation is an $O(\log \log N)$ -approximation, from which the result follows. \square

Along with the SCC bounds proven in Chapter 6, Theorem 7.3.2 implies the existence of an $O(\log \log N)$ -approximation algorithms for many capacitated covering and hitting set problems, including weighted and capacitated versions of \mathbb{R}^3 -**Unit-Cube-SC**, \mathbb{R}^3 -**Unit-Cube-HS**, \mathbb{R}^3 -**3-Sided-Box-SC**, \mathbb{R}^3 -**3-Sided-Box-HS**, \mathbb{R}^3 -**Halfspace-SC**, \mathbb{R}^3 -**Halfspace-HS**, \mathbb{R}^2 -**Disk-SC**, \mathbb{R}^2 -**Disk-HS**, and so on.

7.4 Combinatorial Approximation Algorithm for Priority Tree Cover

Here, we describe a purely combinatorial approach to priority covering on trees. Unfortunately, this method only works in the unweighted case.

Theorem 7.4.1. *There is an efficient 2-approximation algorithm for the unweighted **Priority-Tree-SC** problem.*

The crucial idea is the following. Given an optimum solution $S^* \subseteq \mathcal{S}$, we can partition the edge-set E of T into disjoint sets E_1, \dots, E_p , and partition two copies of S^* into S_1, \dots, S_p , such that E_i is a path in T for each i , and S_i is a priority line cover for the path E_i . Once again, we assume without loss of generality that the instance is segment-complete.

In particular, we prove the following lemma. Let $\hat{E}_{S^*,j}$ be the set of edges e such that j is the segment with the highest supply, among all segments in S^* that cover e . Note that the union of all $\hat{E}_{S^*,j}$, over all $j \in S^*$, partitions E . Also note that for each edge e , there is a unique segment j such that $e \in \hat{E}_{S^*,j}$. If there were two, we could replace one of the segments by a sub-segment and still stay feasible. We call the segment j *responsible* for e .

Lemma 7.4.2. *Given an optimal solution $S^* \subseteq \mathcal{S}$ to a PTC instance with tree $T = (V, E)$, there is a partition*

$$E_1 \cup \dots \cup E_p = E,$$

where each E_i is the edge set of a path in T such that for all $j \in S^*$, $\hat{E}_{S^*,j} \cap E_i \neq \emptyset$ for at most two $i \in \{1, \dots, p\}$.

7.4. COMBINATORIAL APPROXIMATION ALGORITHM FOR PRIORITY TREE COVER

Using this, we describe the 2-approximation algorithm that proves Theorem 7.4.1.

Proof of Theorem 7.4.1. For any two vertices t (top) and b (bottom) of the tree T , such that t is an ancestor of b , let P_{tb} be the unique path from b to t . Note that P_{tb} , together with the restrictions of the segments in \mathcal{S} to P_{tb} , defines an instance of PLC. Therefore, for each pair t and b , we can compute the optimal solution to the corresponding PLC instance; let the cost of this solution be c'_{tb} . Create an instance of the 0,1-tree cover problem with T and segments $\mathcal{S}' := \{(t, b) : t \text{ is an ancestor of } b\}$ with costs c'_{tb} . Solve the 0,1-tree cover instance exactly (recall we are in the rooted version) and for the segments (t, b) in \mathcal{S}' returned, return the solution of the corresponding PLC instance of cost c'_{tb} . We now use Lemma 7.4.2 to obtain a solution to the 0,1-tree cover problem (T, \mathcal{S}') of cost at most 2 times the cost of S^* . This will prove the theorem.

For each E_i , let t_i and b_i be the end points of E_i with t_i being the ancestor of b_i . Since E_i 's partition the edges, the segments $(t_i, b_i) : i = 1, \dots, p$ is a feasible 0,1-tree cover for (T, \mathcal{S}') . Define $S_i := \{j \in S^* : e \in E_i \cap \hat{E}_{S^*,j}\}$ to be the set of segments responsible for the edges in E_i . By definition, S_i is a PLC for E_i . Thus, the cost of the segments in S_i is at least $c'_{t_i b_i}$. Furthermore, Lemma 7.4.2 implies that the total cost of the segments in S_i is at most twice the cost of segments in S^* . Therefore, the cost of the feasible solution to the cover problem in (T, \mathcal{S}') is at most twice the cost of segments in S^* . □

Proof of Lemma 7.4.2. We give an algorithm to compute the decomposition. Let e be any of the edges incident to the root of T , and let $j_1 \in S^*$ be the highest-supply segment covering e . We then let E_1 be the edges of the path in T corresponding to j_1 . Removing E_1 from T yields sub-trees T_1, \dots, T_q . For each tree T_i we repeat the above steps, and let

$$E_1, \dots, E_p \tag{7.4.1}$$

be the final partition; let $j_i \in S^*$ be the segment corresponding to edge-set E_i . Note that for $q < q'$, $\hat{E}_{S^*,j_q} \cap E_{q'}$ is empty. This is because $E_{q'}$ is a subset of edges that are not in $j_{q'-1}, \dots, j_1$.

Consider a segment $j \in S$, and let $1 \leq i \leq p$ be smallest such that $\hat{E}_{S^*,j} \cap E_i \neq \emptyset$, and assume that $\hat{E}_{S^*,j} \cap E_q \neq \emptyset$ for some $i < q \leq p$; choose q smallest with this property. We claim that $j_q = j$, and hence for all $q < q' \leq p$ we have $\hat{E}_{S^*,j} \cap E_{q'} = \emptyset$. Thus, $\hat{E}_{S^*,j}$ has non-empty intersection only with E_i and E_q .

Let $e \in E_{S^*,j} \cap E_i$, and let $f \in E_{S^*,j} \cap E_q$ be two edges in different parts of the partition such that j is responsible for both. As both e and f are edges on j , and since $i < q$, it follows that f is a descendant of e in tree T . Let g be the topmost edge of E_q ; clearly, g is on the e, f -path in T . By the decomposition algorithm, segment j_q is the highest-supply segment covering edge g . As j contains g , this means that the supply of j_q is at least that of j . Finally, since

7. CAPACITATED AND PRIORITY COVERING PROBLEMS

f is on j_q , j_q covers f as well. But this means that $j_q = j$ as j is responsible for f . \square

Chapter 8

Summary and Conclusion

In this thesis, we examined a variety of covering problems, obtaining both algorithmic and hardness results. Motivated by both theoretical and practical applications, we studied geometric problems like \mathbb{R}^2 -**Rectangle-SC**, and combinatorial problems such as **Tree-SC** and its variants. As we have seen, a distinction between computational geometry and combinatorial optimization cannot be clearly drawn, and there are many situations in which methods from both areas can be combined to yield additional insight.

In Chapter 2, we observed that the combinatorial notion of adding *priorities* to a problem has a very useful geometrical interpretation—that of ‘extruding’ the set system out into an extra dimension, with the depth of elements and sets arranged according to supply and demand priorities. We showed how **Tree-SC** can be encoded geometrically using 3-sided rectangles, from which it follows that both **Priority-Tree-SC** and \mathbb{R}^2 -**Rectangle-SC** are subproblems of \mathbb{R}^3 -**4-Sided-Box-SC**. Oddly enough, we obtained an LP-relative constant approximation for weighted **Priority-Tree-SC**, whereas \mathbb{R}^2 -**Rectangle-SC** has no known constant approximation and exhibits an $\Omega(\log M)$ integrality gap, even in the unweighted case (via the recent result of Pach and Tardos [PT11]).

In Chapter 4, we showed that the less general problem \mathbb{R}^2 -**3-Sided-Box-SC** and its hitting set version \mathbb{R}^2 -**3-Sided-Box-HS** both admit polynomial-time exact algorithms via dynamic programming, even in the weighted case. We provided generalizations to more sophisticated set systems, such as families of pseudodisks containing a common point, and families of downward shadows of 2-intersecting functions. In Chapter 5, we ruled out any generalization of this to downward shadows of 3-intersecting functions by developing and encoding a new **APX**-hard problem known as **Special-3SC**. We also provided a variety of other encodings, yielding **APX**-hardness proofs for covering problems involving objects such as half-spaces in \mathbb{R}^4 and ϵ -perturbed copies of a single square or circle in the plane. These results nicely complement the recent PTAS results obtained by Mustafa and Ray [MR10] for problems such as unweighted \mathbb{R}^2 -**Unit-Square-SC**, \mathbb{R}^2 -**Unit-Disk-SC**, and \mathbb{R}^3 -**Halfspace-SC**.

The quasi-uniform sampling algorithm we provided in Chapter 6 improves

upon several known algorithms for geometric covering problems. However, using the notion of *shallow cell complexity*, we were able to generalize this approach to allow nongeometric applications. We provided constant or almost-constant approximations for weighted **Priority-Tree-SC** and a variety of other weighted covering problems exhibiting low *union complexity*, matching the performance of known *unweighted* methods. We also answered an open question of Varadarajan [Var10] by improving several almost-constant approximations to constant ones.

Finally, in Chapter 7, we examined more general covering problems involving capacities and demands. Our key result was a linear programming rounding algorithm, which demonstrated that good approximation algorithms for (weighted) capacitated covering problems could be obtained using algorithms for related (weighted) priority covering and multicover problems as a subroutine.

We hope that our approach has yielded a unified understanding of how the difficulty of a covering problem is tied to its structure. We devote the rest of this section to classifying the problems we have encountered by their complexity, and discussing open problems.

8.1 Summary of Results

Here, we summarize most of the algorithmic and hardness results discussed or proven in this thesis.

The following problems are *totally unimodular* (TUM) and thus admit an exact linear programming formulation (and hence an exact solution in polynomial time, even in the weighted case):

- **\mathbb{R} -Interval-SC.**
- More generally, **Vertical-Tree-SC** and **Network-SC**.
- The dual versions **\mathbb{R} -Interval-HS**, **Vertical-Tree-HS**, and **Network-HS**.

The following problems admit polynomial-time exact algorithms via dynamic programming, even in the weighted case:

- **\mathbb{R}^2 -2-Intersecting-Shadow-SC** (and specifically, **\mathbb{R}^2 -3-Sided-Box-SC**).
- **\mathbb{R}^2 -Origin-Containing-Pseudodisk-SC**.
- **\mathbb{R}^2 -Halfplane-SC** and its self-dual **\mathbb{R}^2 -Halfplane-HS**.
- **\mathbb{R}^2 -3-Sided-Box-HS**.

The following problems are **NP**-hard (even in the unweighted case) but not known (or known not) to be **APX**-hard:

- **\mathbb{R}^2 -Unit-Disk-SC**, **\mathbb{R}^2 -Unit-Disk-DS**, and **\mathbb{R}^2 -Unit-Square-SC**.
- The self-duals **\mathbb{R}^2 -Unit-Square-HS** and **\mathbb{R}^2 -Unit-Disk-HS**.

- \mathbb{R}^3 -**Halfspace-SC** and \mathbb{R}^3 -**Halfspace-HS**.
- **Rectilinear-Polygon-Cover** when the rectilinear polygon has no holes.

The following problems admit a PTAS (only in the unweighted case) via the local search method of Mustafa and Ray:

- \mathbb{R}^3 -**Halfspace-SC** and \mathbb{R}^3 -**Halfspace-HS**.
- \mathbb{R}^2 -**Pseudodisk-HS**, as well as \mathbb{R}^2 -**Disk-SC**, \mathbb{R}^2 -**Disk-HS**, and \mathbb{R}^2 -**Disk-DS**.
- \mathbb{R}^2 -**Unit-Square-SC**, \mathbb{R}^2 -**Unit-Square-DS**, and \mathbb{R}^2 -**Unit-Square-HS** (also in the weighted case via the shifted grids approach of Erlebach and van Leeuwen).

The following problems are known to be **APX**-hard:

- **3-Regular-Graph-SC**.
- **Rectilinear-Polygon-Cover**.
- **Tree-SC**.
- \mathbb{R}^2 -**Fat-Triangle-Cover**.
- \mathbb{R}^2 -**Circle-SC** and \mathbb{R}^3 -**Plane-SC** (covering with *boundaries* of disks and half-planes).

The following problems can be proven **APX**-hard via encodings of **Special-3SC**:

- \mathbb{R}^2 -**Rectangle-SC** and \mathbb{R}^2 -**Ellipse-Cover**, even when all of the objects are ϵ -perturbed copies of a single square or circle.
- \mathbb{R}^2 -**Segment-Shadow-SC**.
- \mathbb{R}^3 -**Unit-Ball-SC** and \mathbb{R}^3 -**Unit-Ball-HS**.
- \mathbb{R}^2 -**Slab-SC**, \mathbb{R}^2 -**Slab-HS**, \mathbb{R}^4 -**Halfspace-SC**, and \mathbb{R}^4 -**Halfspace-HS**.
- \mathbb{R}^2 -**Rectangle-SC** and \mathbb{R}^2 -**Rectangle-HS**, even when each pair of rectangles intersect exactly 0 times or 4 times.
- \mathbb{R}^3 -**Cube-SC**.
- \mathbb{R}^2 -**Fat-Wedge-Cover**.
- **Priority-Vertical-Tree-SC**.

The following problems are known to admit $O(1)$ -factor algorithms:

- **Tree-SC** (a 2-approximation is known in the weighted case, and a 1.5-approximation is known in the unweighted case).

- **Δ -Regular-SC** and **k -Uniform-SC**.

The following problems admit constant or almost-constant approximation algorithms via our quasi-uniform sampling method:

- **Priority-Tree-SC**.
- **\mathbb{R}^2 -Fat-Triangle-Cover**.
- **\mathbb{R}^3 -Unit-Cube-SC** and **\mathbb{R}^3 -Unit-Cube-HS**.
- **\mathbb{R}^2 -Disk-SC**, **\mathbb{R}^2 -Disk-HS**, **\mathbb{R}^3 -Halfspace-SC**, and **\mathbb{R}^3 -Halfspace-HS**.
- More generally, **\mathbb{R}^2 -Pseudodisk-SC** and **\mathbb{R}^2 -Pseudodisk-HS**,
- **\mathbb{R}^3 -3-Sided-Box-SC** and **\mathbb{R}^3 -3-Sided-Box-HS**.

The following problems admit a super-constant integrality gap via the methods of Pach and Tardos:

- **\mathbb{R}^2 -Rectangle-SC** (an $\Omega(\log OPT)$ integrality gap is known, and no approximation of value $o(\log OPT)$ is known).
- **\mathbb{R}^2 -Rectangle-HS** (an $\Omega(\log \log OPT)$ integrality gap is known, and an LP-relative $O(\log \log OPT)$ -approximation is known, but no better approximation is known).

The following problems are equivalent to general **Min-Set-Cover**, have unbounded *VC dimension*, and admit no $o(\log M)$ -approximation unless **P = NP**, even in the unweighted case:

- **Min-Hitting-Set**.
- **Graph-Path-SC**.

8.2 Directions for Future Research

The **\mathbb{R}^2 -Rectangle-SC** and **Tree-SC** problems remain two of the most important and well-studied problems in the area. Despite our progress, there are still countless open questions related to both.

The integrality gap results of Pach and Tardos [PT11] rule out the possibility of an LP-relative constant approximation algorithm for either **\mathbb{R}^2 -Rectangle-SC** or **\mathbb{R}^2 -Rectangle-HS**. However, no super-constant inapproximability has been proven for either problem. In fact, we do not know of *any* natural problem of bounded VC dimension having super-constant inapproximability—given the current state of the art, we believe that it still remains plausible that all covering problems of VC dimension d admit an $f(d)$ -approximation for some function f . However, stronger linear programming relaxations or new methods would be required to prove such a thing. On the other hand, perhaps **\mathbb{R}^2 -Rectangle-SC** or **\mathbb{R}^2 -Rectangle-HS** are simply hard problems. One idea might be to examine

if constructions similar to those in [PT11] could be used alongside methods similar to those in [AMS06] or [Fei98] to prove super-constant hardness for \mathbb{R}^2 -**Rectangle-SC** or \mathbb{R}^2 -**Rectangle-HS**.

One may also note that even an LP-relative $O(1)$ -approximation has still not been ruled out for **Rectilinear-Polygon-Cover**. Erdős’s question concerning the ratio of the maximum independent set to the minimum rectilinear polygon cover appears to remain unanswered. Perhaps some of the more modern methods could lend some fresh insight to this problem.

As for **Tree-SC**, the biggest unanswered question remains its theoretical best possible polynomial-time approximability. The least known upper bound is 1.5 (2 in the weighted case), but the best known lower bound is barely larger than 1, leaving a considerable gap. We suspect that many of the approximation factors implied by our quasi-uniform sampling algorithm in Chapter 6 are likely not optimal either.

There still remain a few key covering problems whose membership in various complexity classes is not known. For example, the local search method of Mustafa and Ray [MR10] fails to yield a PTAS for \mathbb{R}^2 -**Pseudodisk-SC**, and it does not work for weighted problems. Could a weight-sensitive local-search method be developed?

Another question concerns whether the techniques of Chapter 5 can be extended to additional problems. The packing or independent set versions of **Special-3SC** may warrant further study.

Finally, we ask several questions about *shallow cell complexity*—our new matrix parameter. Its connection to the approximability of covering problems makes it a valuable quantity to study, but it currently remains still poorly understood. If one could obtain new structural results linking **SCC** to other matrix properties, then perhaps we could expand the types of covering problems solvable via quasi-uniform sampling and related methods. Can **SCC** be related to statistical properties of a matrix, such a correlation among its rows? Can we find examples of set systems arising in industrial problems or economic models that exhibit low **SCC**? Are there other matrix parameters related to **SCC** that might be more general or easier to use?

We believe that there is ample opportunity for lots of exciting research to arise from these questions. There still remains much to learn concerning the relationship between the approximability and structure of covering problems.

Bibliography

- [AB09] S. Arora and B. Barak, *Complexity theory: A modern approach*, Cambridge University Press, Cambridge, 2009.
- [AEMN06] C. Ambühl, T. Erlebach, M. Mihalák, and M. Nunkesser, *Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs*, Approximation, randomization and combinatorial optimization, Lecture Notes in Comput. Sci., vol. 4110, Springer, Berlin, 2006, pp. 3–14.
- [AES10] B. Aronov, E. Ezra, and M. Sharir, *Small-size ϵ -nets for axis-parallel rectangles and boxes*, SIAM J. Comput. **39** (2010), no. 7, 3248–3282.
- [AK00] P. Alimonti and V. Kann, *Some APX-completeness results for cubic graphs*, Theoret. Comput. Sci. **237** (2000), no. 1-2, 123–134.
- [AKR03] V. S. Anil Kumar and H. Ramesh, *Covering rectilinear polygons with axis-parallel rectangles*, SIAM J. Comput. **32** (2003), no. 6, 1509–1541.
- [ALM⁺98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM **45** (1998), no. 3, 501–555.
- [Alo10] N. Alon, *A non-linear lower bound for planar epsilon-nets*, Proceedings, IEEE Symposium on Foundations of Computer Science, 2010, pp. 341–346.
- [AMS06] N. Alon, D. Moshkovitz, and S. Safra, *Algorithmic construction of sets for k -restrictions*, ACM Trans. Alg. **2** (2006), no. 2, 153–177.
- [ANP⁺04] P. K. Agarwal, E. Nevo, J. Pach, R. Pinchasi, M. Sharir, and S. Smorodinsky, *Lenses in arrangements of pseudo-circles and their applications*, J. ACM **51** (2004), no. 2, 139–186 (electronic).
- [Aur91] F. Aurenhammer, *Voronoi diagrams: A survey of a fundamental geometric data structure*, ACM Comput. Surv. **23** (1991), no. 3, 345–405.

BIBLIOGRAPHY

- [Bal75] E. Balas, *Facets of the knapsack polytope*, Math. Programming **8** (1975), 146–164.
- [BD97] P. Berman and B. DasGupta, *Complexities of efficient solutions of rectilinear polygon cover problems*, Algorithmica **17** (1997), no. 4, 331–356.
- [BD99] H. Brönnimann and O. Devillers, *The union of unit balls has quadratic complexity, even if they all contain the origin*, arXiv eprint, <http://arxiv.org/abs/cs/9907025v1>, 1999.
- [BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Learnability and the Vapnik-Chervonenkis dimension*, J. Assoc. Comput. Mach. **36** (1989), no. 4, 929–965.
- [BG95] H. Brönnimann and M. T. Goodrich, *Almost optimal set covers in finite VC-dimension*, Discrete & Comput. Geom. **14** (1995), no. 4, 463–479.
- [BKS11] N. Bansal, R. Krishnaswamy, and B. Saha, *On capacitated set cover problems*, Proceedings, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2011, To appear.
- [BL73] T. H. Brylawski and D. Lucas, *Uniquely representable combinatorial geometries*, Proceedings, Colloquio Internazionale sulle Teorie Combinatorie, 1973, pp. 83–104.
- [BNBYF⁺01] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, *A unified approach to approximating resource allocation and scheduling*, J. ACM **48** (2001), no. 5, 1069–1090.
- [BP10] N. Bansal and K. Pruhs, *The geometry of scheduling*, Proceedings, IEEE Symposium on Foundations of Computer Science, 2010, pp. 407–414.
- [BSTY98] J.-D. Boissonnat, M. Sharir, B. Tagansky, and M. Yvinec, *Voronoi diagrams in higher dimensions under certain polyhedral distance functions*, Discrete Comput. Geom. **19** (1998), no. 4, 473–484.
- [CCJ90] B. N. Clark, C. J. Colbourn, and D.S. Johnson, *Unit disk graphs*, Discrete Math. **86** (1990), no. 1-3, 165–177.
- [CEK09] C. Chekuri, A. Ene, and N. Korula, *Unsplittable flow in paths and trees and column-restricted packing integer programs*, Proceedings, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2009.

BIBLIOGRAPHY

- [CFLP00] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips, *Strengthening integrality gaps for capacitated network design and covering problems*, Proceedings, ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 106–115.
- [CG11] T. M. Chan and E. Grant, *Exact algorithms and α -hardness results for geometric set cover*, Proceedings, Canadian Conference on Computational Geometry, 2011.
- [CGG06] Y. Chen, M. Grohe, and M. Grüber, *On parameterized approximability*, Parameterized and Exact Computation (Hans Bodlaender and Michael Langston, eds.), Lecture Notes in Computer Science, vol. 4169, Springer Berlin / Heidelberg, 2006, pp. 109–120.
- [CGK10a] D. Chakrabarty, E. Grant, and J. Könemann, *On column-restricted and priority covering integer programs*, Integer Programming and Combinatorial Optimization (Friedrich Eisenbrand and F. Shepherd, eds.), Lecture Notes in Computer Science, vol. 6080, Springer Berlin / Heidelberg, 2010, pp. 355–368.
- [CGK10b] D. Chakrabarty, E. Grant, and J. Könemann, *On column-restricted and priority covering integer programs*, arXiv eprint, <http://arxiv.org/abs/1003.1507>, 2010.
- [CGKS12] T. M. Chan, E. Grant, J. Könemann, and M. Sharpe, *Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling*, Proceedings, ACM-SIAM Symposium on Discrete Algorithms, 2012, To appear.
- [CGNS08] J. Chuzhoy, A. Gupta, J. Naor, and A. Sinha, *On the approximability of some network design problems*, ACM Trans. Alg. **4** (2008), no. 2, Art. 23, 17.
- [Cha03] T. M. Chan, *Polynomial-time approximation schemes for packing and piercing fat objects*, J. Algorithms **46** (2003), no. 2, 178–189.
- [Chv79] V. Chvátal, *A greedy heuristic for the set covering problem*, Math. Oper. Res. **4** (1979), 233–235.
- [CIK88] Y. Cheng, S.S. Iyengar, and R.L. Kashyap, *A new method for image compression using irreducible covers of maximal rectangles*, IEEE Trans. Software Engrg. **14** (1988), 651–658.
- [CJR99] J. Cheriyan, T. Jordán, and R. Ravi, *On 2-coverings and 2-packings of laminar families*, Proceedings, European Symposium on Algorithms, 1999, pp. 510–520.
- [CKSS81] S. Chaiken, D. J. Kleitman, M. Saks, and J. Shearer, *Covering regions by rectangles*, SIAM J. Algebraic Discrete Methods **2** (1981), no. 4, 394–410.

BIBLIOGRAPHY

- [CMS07] C. Chekuri, M. Mydlarz, and F. B. Shepherd, *Multicommodity demand flow in a tree and packing integer programs*, ACM Trans. Algorithms **3** (2007), no. 3, Art. 27, 23.
- [CNS04] M. Charikar, J. Naor, and B. Schieber, *Resource optimization in QoS multicast routing of real-time multimedia*, IEEE/ACM Trans. Netw. **12** (2004), no. 2, 340–348.
- [Cox69] H. S. M. Coxeter, *Introduction to geometry*, second ed., John Wiley & Sons Inc., New York, 1969.
- [CP91] P. Crescenzi and A. Panconesi, *Completeness in approximation classes*, Inform. and Comput. **93** (1991), no. 2, 241–262.
- [CR94] J. C. Culberson and R. A. Reckhow, *Covering polygons is hard*, J. Algorithms **17** (1994), no. 1, 2–44.
- [CS89] K. L. Clarkson and P. W. Shor, *Applications of random sampling in computational geometry, II*, Discrete & Comput. Geom. **4** (1989), no. 1, 387–421.
- [CT00] P. Crescenzi and L. Trevisan, *On approximation scheme preserving reducibility and its applications*, Theory Comput. Syst. **33** (2000), no. 1, 1–16.
- [CV07] K. L. Clarkson and K. R. Varadarajan, *Improved approximation algorithms for geometric set cover*, Discrete & Comput. Geom. **37** (2007), no. 1, 43–58.
- [dB10] Mark de Berg, *Better bounds on the union complexity of locally fat objects*, Computational geometry (SCG’10), ACM, New York, 2010, pp. 39–47.
- [dBCvKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational geometry*, third ed., Springer-Verlag, Berlin, 2008, Algorithms and applications.
- [Dob82] G. Dobson, *Worst-case analysis of greedy heuristics for integer programming with non-negative data*, Math. Oper. Res. **7** (1982), no. 4, 515–531.
- [DS05] I. Dinur and S. Safra, *On the hardness of approximating minimum vertex cover*, Ann. of Math. (2) **162** (2005), no. 1, 439–485.
- [DSW94] G. Ding, P. Seymour, and P. Winkler, *Bounding the vertex cover number of a hypergraph*, Combinatorica **14** (1994), no. 1, 23–34.
- [Dud78] R. M. Dudley, *Central limit theorems for empirical measures*, Ann. Probab. **6** (1978), no. 6, 899–929 (1979).

BIBLIOGRAPHY

- [EAS11] E. Ezra, B. Aronov, and M. Sharir, *Improved bound for the union of fat triangles*, Proceedings, ACM-SIAM Symposium on Discrete Algorithms, 2011, pp. 1778–1785.
- [EFKN01] G. Even, J. Feldman, G. Kortsarz, and Z. Nutov, *A 3/2-approximation algorithm for augmenting the edge-connectivity of a graph from 1 to 2 using a subset of a given edge set (extended abstract)*, Approximation, randomization, and combinatorial optimization (Berkeley, CA, 2001), Lecture Notes in Comput. Sci., vol. 2129, Springer, Berlin, 2001, pp. 90–101.
- [ERS93] A. Efrat, G. Rote, and M. Sharir, *On the union of fat wedges and separating a collection of segments by a line*, Comput. Geom. **3** (1993), no. 5, 277–288.
- [ERS05] G. Even, D. Rawitz, and S. Shahar, *Hitting sets when the VC-dimension is small*, Inform. Process. Lett. **95** (2005), no. 2, 358–362.
- [EvL10] T. Erlebach and E. J. van Leeuwen, *PTAS for weighted set cover on unit squares*, APPROX and RANDOM (2010), 166–177.
- [Fei98] U. Feige, *A threshold of $\ln n$ for approximating set cover*, J. ACM **45** (1998), no. 4, 634–652.
- [FJ82] G. N. Frederickson and J. Ja’Ja’, *On the relationship between the biconnectivity augmentation and traveling salesman problems*, Theoret. Comput. Sci. **19** (1982), no. 2, 189–201.
- [FK84] D. S. Franzblau and D. J. Kleitman, *An algorithm for covering polygons with rectangles*, Inform. and Control **63** (1984), no. 3, 164–189.
- [FPT81] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, *Optimal packing and covering in the plane are NP-complete*, Inform. Process. Lett. **12** (1981), no. 3, 133–137.
- [Fra89] D. S. Franzblau, *Performance guarantees on a sweep-line heuristic for covering rectilinear polygons with rectangles*, SIAM J. Discrete Math. **2** (1989), no. 3, 307–321.
- [GP10] M. Gibson and I. A. Pirwani, *Algorithms for dominating set in disk graphs: Breaking the $\log n$ barrier*, Proceedings, European Symposium on Algorithms, 2010, pp. 243–254.
- [GRV05] C. Glaßer, S. Reith, and H. Vollmer, *The complexity of base station positioning in cellular networks*, Discrete Appl. Math. **148** (2005), no. 1, 1–12.

BIBLIOGRAPHY

- [Hås99] J. Håstad, *Some optimal inapproximability results*, STOC '97 (El Paso, TX), ACM, New York, 1999, pp. 1–10 (electronic).
- [Heg82] A. Hegedüs, *Algorithms for covering polygons with rectangles*, Comput. Aided Geom. Design **14** (1982), 257–260.
- [HHL02] S. Hannenhalli, E. Hubell, R. Lipshutz, and P. Pevzner, *Combinatorial algorithms for design of DNA arrays*, Adv. Biochem. Eng. Biotechnol. **77** (2002), 1–19.
- [HJP75] P. Hammer, E. Johnson, and U. Peled, *Facets of regular 0-1 polytopes*, Math. Programming **8** (1975), 179–206.
- [HLL06] L. Heinrich-Litan and M. E. Lübbecke, *Rectangle covers revisited computationally*, ACM J. Exp. Algorithmics **11** (2006), 21 pp. (electronic).
- [HM85] D. S. Hochbaum and W. Maass, *Approximation schemes for covering and packing problems in image processing and VLSI*, J. Assoc. Comput. Mach. **32** (1985), no. 1, 130–136.
- [Hoc82] D. S. Hochbaum, *Approximation algorithms for the set covering and vertex cover problems*, SIAM J. Comput. **11** (1982), no. 3, 555–556.
- [HP09] S. Har-Peled, *Being fat and friendly is not enough*, arXiv eprint, <http://arxiv.org/abs/0908.2369v1>, 2009.
- [HPL08] S. Har-Peled and M. Lee, *Weighted geometric set cover problems revisited*, Unpublished manuscript, http://valis.cs.uiuc.edu/~sariel/papers/08/expand_cover/expand_cover.pdf, 2008.
- [HW87] D. Haussler and E. Welzl, *ϵ -nets and simplex range queries*, Discrete & Comput. Geom. **2** (1987), 127–151.
- [Jae79] F. Jaeger, *Flows and generalized coloring theorems in graphs*, J. Combin. Theory Ser. B **26** (1979), 205–216.
- [Jai01] K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica **21** (2001), no. 1, 39–60.
- [JM02] D. S. Johnson and L. A. McGeoch, *Experimental analysis of heuristics for the tsp*, The Traveling Salesman Problem and its Variations (Gregory Gutin and Abraham P. Punnen, eds.), Kluwer Academic Publishers, Dordrecht, 2002.
- [Jof74] A. Joffe, *On a set of almost deterministic k -independent random variables*, Ann. Probab. **2** (1974), no. 1, 161–162.

BIBLIOGRAPHY

- [Joh74] D. S. Johnson, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci. **9** (1974), 256–278.
- [Kai97] T. Kaiser, *Transversals of d -intervals*, Discrete Comput. Geom. **18** (1997), no. 2, 195–203.
- [Kan92] V. Kann, *On the approximability of NP-complete optimization problems*, Ph.D. thesis, Department of Numerical Analysis and Computing, Royal Institute of Technology, Stockholm, 1992.
- [Kar72] R. M. Karp, *Reducibility among combinatorial problems*, Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), Plenum, New York, 1972, pp. 85–103.
- [Kes07] B. Keszegh, *Weak conflict-free colorings of point sets and simple regions*, In Proc. Canad. Conf. on Computational Geometry, 2007, pp. 97–100.
- [Khu97] S. Khuller, *Approximation algorithms for finding highly connected subgraphs*, Approximation Algorithms for NP-hard Problems (D. S. Hochbaum, ed.), PWS Publishing, 1997.
- [KLPS86] K. Kedem, R. Livne, J. Pach, and M. Sharir, *On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles*, Discrete & Comput. Geom. **1** (1986), 59–70.
- [KN07] G. Kortsarz and Z. Nutov, *Approximating minimum cost connectivity problems*, Handbook of approximation algorithms and metaheuristics (Teofilo F. Gonzalez, ed.), Chapman & Hall/CRC Computer and Information Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2007.
- [Kol03] S. G. Kolliopoulos, *Approximating covering integer programs with multiplicity constraints*, Discrete Appl. Math. **129** (2003), no. 2-3, 461–473.
- [Kor09] N. Korula, private communication, 2009.
- [KPW92] J. Komlós, J. Pach, and G. J. Woeginger, *Almost tight bounds for epsilon-nets*, Discrete & Comput. Geom. **7** (1992), 163–173.
- [KR08] S. Khot and O. Regev, *Vertex cover might be hard to approximate to within $2 - \epsilon$* , J. Comput. System Sci. **74** (2008), no. 3, 335–349.
- [KS01] S. G. Kolliopoulos and C. Stein, *Approximation algorithms for single-source unsplittable flow*, SIAM J. Comput. **31** (2001), no. 3, 919–946.

BIBLIOGRAPHY

- [KS04] S. G. Kolliopoulos and C. Stein, *Approximating disjoint-path problems using packing integer programs*, Math. Programming **99** (2004), no. 1, 63–87.
- [KY05] S. G. Kolliopoulos and N. E. Young, *Approximation algorithms for covering/packing integer programs*, J. Comput. System Sci. **71** (2005), no. 4, 495–505.
- [LMR91] N. Linial, Y. Mansour, and R. L. Rivest, *Results on learnability and the Vapnik-Chervonenkis dimension*, Inform. and Comput. **90** (1991), no. 1, 33–49.
- [Lon01] P. Long, *Using the pseudo-dimension to analyze approximation algorithms for integer programming*, Algorithms and Data Structures (Frank Dehne, Jrg-Rdiger Sack, and Roberto Tamassia, eds.), Lecture Notes in Computer Science, vol. 2125, Springer Berlin / Heidelberg, 2001, pp. 26–37.
- [Lov75] L. Lovász, *On the ratio of optimal integral and fractional covers*, J. of Discrete Mathematics **13** (1975), 383–390.
- [Lub85] A. Lubiw, *Ordering and some combinatorial optimization problems with some geometric applications*, Ph.D. thesis, University of Waterloo, 1985.
- [Lub90] A. Lubiw, *The Boolean basis problem and how to cover some polygons by rectangles*, SIAM J. Discrete Math. **3** (1990), no. 1, 98–115.
- [LY94] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, J. Assoc. Comput. Mach. **41** (1994), no. 5, 960–981.
- [Mad09] Y. Maduel, *Covering a laminar family by leaf to leaf links*, Master’s thesis, The Open University of Israel, 2009.
- [Mat92] J. Matoušek, *Reporting points in halfspaces*, Comput. Geom. **2** (1992), no. 3, 169–186.
- [Mat03] J. Matoušek, *Using the Borsuk-Ulam theorem*, Universitext, Springer-Verlag, Berlin, 2003, Lectures on topological methods in combinatorics and geometry, Written in cooperation with Anders Björner and Günter M. Ziegler.
- [MM99] T. A. McKee and F. R. McMorris, *Topics in intersection graph theory*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

BIBLIOGRAPHY

- [MPS⁺94] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl, *Fat triangles determine linearly many holes*, SIAM J. Comput. **23** (1994), no. 1, 154–169.
- [MR95] R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge University Press, New York, NY, 1995.
- [MR10] N. H. Mustafa and S. Ray, *Improved results on geometric hitting set problems*, Discrete Comput. Geom. **44** (2010), no. 4, 883–895.
- [MSW90] J. Matoušek, R. Seidel, and E. Welzl, *How to net a lot with little: Small ϵ -nets for disks and halfspaces*, In Proc. 6th Annu. ACM Sympos. Comput. Geom, 1990, pp. 16–22.
- [MU02] Elchanan Mossel and Christopher Umans, *On the complexity of approximating the VC dimension*, J. Comput. System Sci. **65** (2002), no. 4, 660–671, Special issue on complexity, 2001 (Chicago, IL).
- [Nag03] H. Nagamochi, *An approximation for finding a smallest 2-edge-connected subgraph containing a specified spanning tree*, Discrete Appl. Math. **126** (2003), no. 1, 83–113, Fifth Annual International Computing and Combinatorics Conference (COCOON’99) (Tokyo).
- [Oxl11] J. Oxley, *Matroid theory*, Oxford University Press, New York, NY, 2011.
- [PR08] E. Pyrga and S. Ray, *New existence proofs for ϵ -nets*, Computational geometry (SCG’08), ACM, New York, 2008, pp. 199–207.
- [PT11] J. Pach and G. Tardos, *Tight lower bounds for the size of epsilon-nets*, Proc. 27th ACM Sympos. Comput. Geom., 2011, pp. 458–463.
- [PY91] C. H. Papadimitriou and M. Yannakakis, *Optimization, approximation, and complexity classes*, J. Comput. System Sci. **43** (1991), no. 3, 425–440.
- [PY96] Christos H. Papadimitriou and Mihalis Yannakakis, *On limited nondeterminism and the complexity of the V-C dimension*, J. Comput. System Sci. **53** (1996), no. 2, part 1, 161–170, Eighth Annual Structure in Complexity Theory Conference (San Diego, CA, 1993).
- [RS97] R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, Proceedings, ACM Symposium on Theory of Computing, 1997, pp. 475–484.

BIBLIOGRAPHY

- [RV93] S. Rajagopalan and V. V. Vazirani, *Primal-dual RNC approximation algorithms for (multi)set (multi)cover and covering integer programs*, Proceedings, IEEE Symposium on Foundations of Computer Science, 1993.
- [SA95] M. Sharir and P. K. Agarwal, *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press, Cambridge, 1995.
- [Sch86] A. Schrijver, *Theory of linear and integer programming*, Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons Ltd., Chichester, 1986, A Wiley-Interscience Publication.
- [Sch03] A. Schrijver, *Combinatorial optimization*, Springer, New York, 2003.
- [Sey80] P. D. Seymour, *Decomposition of regular matroids*, J. Combin. Theory Ser. B **28** (1980), no. 3, 305–359.
- [Sri99] A. Srinivasan, *Improved approximation guarantees for packing and covering integer programs*, SIAM J. Comput. **29** (1999), no. 2, 648–670.
- [Sri06] A. Srinivasan, *An extension of the Lovász local lemma, and its applications to integer programming*, SIAM J. Comput. **36** (2006), no. 3, 609–634.
- [SSS95] J. P. Schmidt, A. Siegel, and A. Srinivasan, *Chernoff-Hoeffding bounds for applications with limited independence*, SIAM J. Discrete Math. **8(2)** (1995), 223–250.
- [Tar95] G. Tardos, *Transversals of 2-intervals, a topological approach*, Combinatorica **15** (1995), no. 1, 123–134.
- [Tre01] L. Trevisan, *Non-approximability results for optimization problems on bounded degree instances*, Proceedings, ACM Symposium on Theory of Computing, 2001, pp. 453–461.
- [Tru92] K. Truemper, *Matroid decomposition*, Academic Press Inc., Boston, MA, 1992.
- [Vap98] V. N. Vapnik, *Statistical learning theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, John Wiley & Sons Inc., New York, 1998.
- [Var09] K. R. Varadarajan, *Epsilon nets and union complexity*, Proceedings, ACM-SIAM Symposium on Computational Geometry, 2009, pp. 11–16.

BIBLIOGRAPHY

- [Var10] K. R. Varadarajan, *Weighted geometric set cover via quasi-uniform sampling*, Proceedings, ACM Symposium on Theory of Computing, 2010, pp. 641–648.
- [Vaz01] V. V. Vazirani, *Approximation algorithms*, Springer-Verlag, Berlin, 2001.
- [VC71] V. N. Vapnik and A. Y. Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*, Theory of Probab. and its Applications **16** (1971), no. 2, 264–280.
- [vL09] E. J. van Leeuwen, *Optimization and approximation on systems of geometric objects*, Ph.D. thesis, Universiteit van Amsterdam, 2009.
- [Wol75] L. Wolsey, *Facets for a linear inequality in 0-1 variables*, Math. Programming **8** (1975), 168–175.