# New Geometric Techniques for Linear Programming and Graph Partitioning

by

Jonathan A. Kelner

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2006





Author .....
Department of Electrical Engineering and Computer Science
August 15, 2006

Certified by .....
Daniel A. Spielman
Professor of Applied Mathematics and Computer Science, Yale University
Thesis Co-Supervisor

Certified by .....
Madhu Sudan
Professor of Computer Science
Thesis Co-Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Committee for Graduate Students

# New Geometric Techniques for Linear Programming and Graph Partitioning

by

Jonathan A. Kelner

Submitted to the Department of Electrical Engineering and Computer Science
on August 15, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

## Abstract

In this thesis, we advance a collection of new geometric techniques for the analysis of combinatorial algorithms. Using these techniques, we resolve several longstanding questions in the theory of linear programming, polytope theory, spectral graph theory, and graph partitioning.

The thesis consists of two main parts. In the first part, which is joint work with Daniel Spielman, we present the first randomized polynomial-time simplex algorithm for linear programming, answering a question that has been open for over fifty years. Like the other known polynomial-time algorithms for linear programming, its running time depends polynomially on the number of bits used to represent its input.

To do this, we begin by reducing the input linear program to a special form in which we merely need to certify boundedness of the linear program. As boundedness does not depend upon the right-hand-side vector, we run a modified version of the shadow-vertex simplex method in which we start with a random right-hand-side vector and then modify this vector during the course of the algorithm. This allows us to avoid bounding the diameter of the original polytope.

Our analysis rests on a geometric statement of independent interest: given a polytope $\{\boldsymbol{x} \mid A\boldsymbol{x} \leq \boldsymbol{b}\}$ in isotropic position, if one makes a polynomially small perturbation to $\boldsymbol{b}$ then the number of edges of the projection of the perturbed polytope onto a random 2-dimensional subspace is expected to be polynomial.

In the second part of the thesis, we address two long-open questions about finding good separators in graphs of bounded genus and degree:

1. It is a classical result of Gilbert, Hutchinson, and Tarjan [25] that one can find asymptotically optimal separators on these graphs if he is given both the graph *and* an embedding of it onto a low genus surface. Does there exist a simple, efficient algorithm to find these separators given only the graph and not the embedding?

2. In practice, spectral partitioning heuristics work extremely well on these graphs. Is there a theoretical reason why this should be the case?

We resolve these two questions by showing that a simple spectral algorithm finds separators of cut ratio $O(\sqrt{g/n})$ and vertex bisectors of size $O(\sqrt{gn})$ in these graphs, both of which are optimal. As our main technical lemma, we prove an $O(g/n)$ bound on the second smallest eigenvalue of the Laplacian of such graphs and show that this is tight, thereby resolving a conjecture of Spielman and Teng. While this lemma is essentially combinatorial in nature, its proof comes from continuous mathematics, drawing on the theory of circle packings and the geometry of compact Riemann surfaces.

While the questions addressed in the two parts of the thesis are quite different, we show that a common methodology runs through their solutions. We believe that this methodology provides a powerful approach to the analysis of algorithms that will prove useful in a variety of broader contexts.

Thesis Co-Supervisor: Daniel A. Spielman
Title: Professor of Applied Mathematics and Computer Science, Yale University

Thesis Co-Supervisor: Madhu Sudan
Title: Professor of Computer Science

# Acknowledgments

First and foremost, I would like to thank Daniel Spielman for being a wonderful mentor and friend. He has taught me a tremendous amount, not just about how to solve problems, but about how to choose them. He helped me at essentially every stage of this thesis, and it certainly would not have been possible without him. I was truly lucky to have him as my advisor, and I owe him a great debt of gratitude.

I would also like to thank:

Sophie, my family, and all of my friends for their continued love, support, and, of course, tolerance of me throughout my graduate career;

Shang-Hua Teng for introducing me to the question addressed in the second part of this thesis and for several extremely helpful conversations about both spectral partitioning and academic life;

All of the faculty and other students in the Theory group at MIT for providing an exciting and stimulating environment for the past four years;

Nathan Dunfield, Phil Bowers, and Ken Stephenson for their guidance on the theory of circle packings

Christopher Mihelich for his help in simplifying the proof of Lemma /citeorthog to ones and for sharing his uncanny knowledge of TeX;

The anonymous referees at the SIAM Journal on Computing for their very helpful comments and suggestions about the spectral partitioning portion of the thesis;

Microsoft Research for hosting me over the summer of 2004;

Madhu Sudan and Piotr Indyk for serving on my thesis committee and for several very helpful comments about how to improve this thesis.

# Previous Publications of this Work

Much of the material in this thesis has appeared in previously published work. In particular, the material on the simplex method in the introduction and the contents of chapters three through seven represent joint work with Daniel Spielman and were presented at STOC 2006 [33]. The introductory material on spectral partitioning and chapters eight through twelve were presented at STOC 2004 [34] and were expanded upon in my Master's thesis [35] and in the STOC 2004 Special Issue of the SIAM Journal on Computing [36]. In all of these cases, this thesis draws from the above works without further mention.

# Contents

**Acknowledgements** 4

**Previous Publications of this Work** 5

**Table of Contents** 6

**List of Figures** 8

**1 Introduction** 9
- 1.1 A Randomized Polynomial-Time Simplex Method for Linear Programming . . . 9
- 1.2 Spectral Partitioning, Eigenvalue Bounds, and Circle Packings for Graphs of Bounded Genus . . . 12
- 1.3 The Common Methodology . . . 13

**I A Randomized Polynomial-Time Simplex Algorithm for Linear Programming** 14

**2 Introduction to Linear Programming Geometry** 15
- 2.1 Linear Programs as Polytopes . . . 15
- 2.2 Duality . . . 16
- 2.3 Polarity . . . 19
- 2.4 When is a Linear Program Unbounded? . . . 19

**3 The Simplex Algorithm** 21
- 3.1 The General Method . . . 21
- 3.2 The Shadow-Vertex Method . . . 22

**4 Bounding the Shadow Size** 23
- 4.1 The Shadow Size in the $k$-Round Case . . . 23
- 4.2 The Shadow Size in the General Case . . . 30

**5 Reduction of Linear Programming to Certifying Boundedness** **34**
5.1 Reduction to a Feasibility Problem . . . . . 34
5.2 Degeneracy and the Reduction to Certifying Boundedness . . . . . 37

**6 Our Algorithm** **41**
6.1 Constructing a Starting Vertex . . . . . 42
6.2 Polytopes that are not $k$-Round . . . . . 43
6.3 Towards a Strongly Polynomial-Time Algorithm for Linear Programming? . . . . . 47

**7 Geometric Lemmas for Algorithm's Correctness** **48**
7.1 2-Dimensional Geometry Lemma . . . . . 49
7.2 High-Dimensional Geometry Lemma . . . . . 50

**II Spectral Partitioning, Eigenvalue Bounds, and Circle Packings for Graphs of Bounded Genus** **52**

**8 Background in Graph Theory and Spectral Partitioning** **53**
8.1 Graph Theory Definitions . . . . . 53
8.2 Spectral Partitioning . . . . . 54

**9 Outline of the Proof of the Main Technical Result** **56**

**10 Introduction to Circle Packings** **58**
10.1 Planar Circle Packings . . . . . 58
10.2 A Very Brief Introduction to Riemann Surface Theory . . . . . 60
10.3 Circle Packings on Surfaces of Arbitrary Genus . . . . . 62

**11 An Eigenvalue Bound** **64**

**12 The Proof of the Subdivision Lemma** **70**

**Bibliography** **79**

# List of Figures


4-1 The points $x$, $y$ and $q$. . . . . 28

7-1 The geometric objects considered in Lemma 7.1.1 . . . . . 49

8-1 The surfaces of genus 0, 1, 2, and 3. . . . . 54

10-1 A univalent circle packing with its associated graph. . . . . 59
10-2 A nonunivalent circle packing with its associated graph. . . . . 59

11-1 The hexagonal subdivision procedure applied to a triangulation with two triangles. . . . . 66

12-1 A subdivided graph, with $P(w)$ and $N(w)$ shaded for a vertex $w$. . . 73
12-2 An illustration of how the grid graph exists as a subgraph of the union of two adjacent subdivided triangles. . . . . 74
12-3 The entire construction illustrated for a given edge of the original graph. 75

# Chapter 1

# Introduction

In this thesis, we advance a collection of new geometric techniques for the analysis of combinatorial algorithms. Using these techniques, we resolve several longstanding questions in the theory of linear programming, polytope theory, spectral graph theory, and graph partitioning.

In this chapter, we shall introduce and summarize the main contributions of this thesis. The remainder of this document will be divided into two main parts. Here, we discuss the main components of each part and then briefly explain the common methodology that runs between the two. The contents of this thesis are drawn heavily from previously published works; please see page 5 for a full discussion of the origins of the different sections.

## 1.1 A Randomized Polynomial-Time Simplex Method for Linear Programming

In the first part of this thesis, we shall present the first randomized polynomial-time simplex method for linear programming. Linear programming is one of the fundamental problems of optimization. Since Dantzig [14] introduced the simplex method for solving linear programs, linear programming has been applied in a diverse range of fields including economics, operations research, and combinatorial optimization. From a theoretical standpoint, the study of linear programming has motivated major advances in the study of polytopes, convex geometry, combinatorics, and complexity theory.

While the simplex method was the first practically useful approach to solving linear programs and is still one of the most popular, it was unknown whether any variant of the simplex method could be shown to run in polynomial time in the worst case. In fact, most common variants have been shown to have exponential worst-case

complexity. In contrast, algorithms have been developed for solving linear programs that do have polynomial worst-case complexity [38, 32, 19, 4]. Most notable among these have been the ellipsoid method [38] and various interior-point methods [32]. All previous polynomial-time algorithms for linear programming of which we are aware differ from simplex methods in that they are fundamentally geometric algorithms: they work either by moving points inside the feasible set, or by enclosing the feasible set in an ellipse. Simplex methods, on the other hand, walk along the vertices and edges defined by the constraints. The question of whether such an algorithm can be designed to run in polynomial time has been open for over fifty years.

We recall that a linear program is a constrained optimization problem of the form:

$$\begin{aligned} &\textbf{maximize} && \boldsymbol{c} \cdot \boldsymbol{x} \\ &\textbf{subject to} && A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{R}^d, \end{aligned} \tag{1.1}$$

where $\boldsymbol{c} \in \mathbb{R}^d$ and $\boldsymbol{b} \in \mathbb{R}^n$ are column vectors, and $A$ is an $n \times d$ matrix. The vector $\boldsymbol{c}$ is the *objective function*, and the set $P := \{\boldsymbol{x} \mid A\boldsymbol{x} \leq \boldsymbol{b}\}$ is the set of *feasible* points. If it is non-empty, $P$ is a convex polyhedron, and each of its extreme vertices will be determined by $d$ constraints of the form $\boldsymbol{a}_i \cdot \boldsymbol{x} = b_i$, where $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ are the rows of $A$. It is not difficult to show that the objective function is always maximized at an extreme vertex, if this maximum is finite.

The first simplex methods used heuristics to guide a walk on the graph of vertices and edges of $P$ in search of one that maximizes the objective function. In order to show that any such method runs in worst-case polynomial time, one must prove a polynomial upper bound on the diameter of polytope graphs. Unfortunately, the existence of such a bound is a wide-open question: the famous Hirsch Conjecture asserts that the graph of vertices and edges of $P$ has diameter at most $n - d$, whereas the best known bound for this diameter is superpolynomial in $n$ and $d$ [31].

Later simplex methods, such as the self-dual simplex method and the criss-cross method [15, 22], tried to avoid this obstacle by considering more general graphs for which better diameter bounds were possible. However, even though some of these graphs have polynomial diameters, they have exponentially many vertices, and nobody had been able to design a polynomial-time algorithm that provably finds the optimum after following a polynomial number of edges. In fact, essentially every such deterministic algorithm has well-known counterexamples on which the walk takes exponentially many steps. However, for randomized pivot rules very little is known. While the best previously known upper bounds on the running time of randomized pivot rule is $\Omega\left(\exp\left(\sqrt{d \log n}\right)\right)$ [30], there exist very simple randomized pivots rules for which essentially no nontrivial lower bounds have been shown.

In this thesis, we present the first randomized polynomial-time simplex method. Like the other known polynomial-time algorithms for linear programming, the running

time of our algorithm depends polynomially on the bit-length of the input. We do *not* prove an upper bound on the diameter of polytopes. Rather we reduce the linear programming problem to the problem of determining whether a set of linear constraints defines an unbounded polyhedron. We then randomly perturb the right-hand sides of these constraints, observing that this does not change the answer, and we then use a shadow-vertex simplex method to try solve the perturbed problem. When the shadow-vertex method fails, it suggests a way to alter the distributions of the perturbations, after which we apply the method again. We prove that the number of iterations of this loop is polynomial with high probability.

It is important to note that the vertices considered during the course of the algorithm may not all appear on a single polytope. Rather, they may be viewed as appearing on the convex hulls of polytopes with different $\boldsymbol{b}$-vectors. It is well-known that the graph of all of these "potential" vertices has small diameter. However, there was previously no way to guide a walk among these potential vertices to one optimizing any particular objective function. Our algorithm uses the graphs of polytopes "near" $P$ to impose structure on this graph and to help to guide our walk.

Perhaps the message to take away from this is that instead of worrying about the combinatorics of the natural polytope $P$, one can reduce the linear programming problem to one whose polytope is more tractable. The first result of this part of the thesis, and the inspiration for the algorithm, captures this idea by showing that if one slightly perturbs the $\boldsymbol{b}$-vector of a polytope in near-isotropic position, then there will be a polynomial-step path from the vertex minimizing to the vertex maximizing a random objective function. Moreover, this path may be found by the shadow-vertex simplex method.

We stress that while our algorithm involves a perturbation, it is intrinsically different from previous papers that have provided average-case or smoothed analyses of linear programming. In those papers, one shows that, given some linear program, one can probably use the simplex method to solve a nearby but different linear program; the perturbation actually modified the input. In the present document, our perturbation is used to inform the walk that we take on the (feasible or infeasible) vertices of our linear program; however, we actually solve the exact instance that we are given. We believe that ours is the first simplex algorithm to achieve this, and we hope that our results will be a useful step on the path to a strongly polynomial-time algorithm for linear programming.

## 1.2 Spectral Partitioning, Eigenvalue Bounds, and Circle Packings for Graphs of Bounded Genus

In the second part of the thesis, we shall take up several long-open problems in the spectral and algorithmic theory of graphs. Spectral methods have long been used as a heuristic in graph partitioning. They have had tremendous experimental and practical success in a wide variety of scientific and numerical applications, including mapping finite element calculations on parallel machines [46, 51], solving sparse linear systems [9, 10], partitioning for domain decomposition, and VLSI circuit design and simulation [8, 28, 2]. However, it is only recently that people have begun to supply formal justification for their efficacy [27, 47]. In [47], Spielman and Teng used the results of Mihail [41] to show that the quality of the partition produced by the application of a certain spectral algorithm to a graph can be established by proving an upper bound on the Fiedler value of the graph (*i.e.*, the second smallest eigenvalue of its Laplacian). They then provided an $O(1/n)$ bound on the Fielder value of a *planar* graph with $n$ vertices and bounded maximum degree. This showed that spectral methods can produce a cut of ratio $O(\sqrt{1/n})$ and a vertex bisector of size $O(\sqrt{n})$ in a bounded degree planar graph.

In this part of the thesis, we use the theory of circle packings and conformal mappings of compact Riemann surfaces to generalize these results to graphs of positive genus. We prove that the Fiedler value of a genus $g$ graph of bounded degree is $O(g/n)$ and demonstrate that this is asymptotically tight, thereby resolving a conjecture of Spielman and Teng. We then apply this result to obtain a spectral partitioning algorithm that finds separators whose cut ratios are $O(\sqrt{g/n})$ and vertex bisectors of size $O(\sqrt{gn})$, both of which are optimal. To our knowledge, this provides the only truly practical algorithm for finding such separators and vertex bisectors for graphs of bounded genus and degree. While there exist other asymptotically fast algorithms for this, they all rely on being given an embedding of the graph in a genus $g$ surface (e.g., [25]). It is not always the case that we are given such an embedding, and computing it is quite difficult. (In particular, computing the genus of a graph is NP-hard [49], and the best known algorithms for constructing such an embedding are either $n^{O(g)}$ [20] or polynomial in $n$ but doubly exponential in $g$ [17]. Mohar has found an algorithm that depends only linearly on $n$ [42], but it has an uncalculated and very large dependence on $g$.) The excluded minor algorithm of Alon, Seymour, and Thomas [1] does not require an embedding of the graph, but the separators that it produces are not asymptotically optimal.

The question of whether there exists an efficient algorithm for providing asymptotically optimal cuts without such an embedding was first posed twenty years ago

by Gilbert, Hutchinson, and Tarjan [25].[1] We resolve this question here, as our algorithm proceeds without any knowledge of an embedding of the graph, and it instead relies only on simple matrix manipulations of the adjacency matrix of the graph. While the analysis of the algorithm requires some somewhat involved mathematics, the algorithm itself is quite simple, and it can be implemented in just a few lines of Matlab code. In fact, it is only a slight modification of the spectral heuristics for graph partitioning that are widely deployed in practice without any theoretical guarantees.

We believe that the techniques that we employ to obtain our eigenvalue bounds are of independent interest. To prove these bounds, we make what is perhaps the first real use of the theory of circle packings and conformal mappings of positive genus Riemann surfaces in the computer science literature. This is a powerful theory, and we believe that it will be useful for addressing other questions in spectral and topological graph theory.

## 1.3 The Common Methodology

While the results proven in the two parts of the thesis are quite different, it will become clear that a common methodology that runs between them. In both cases, we provide bounds on the performance of algorithms using very similar geometric techniques. In particular, the innovations of this thesis revolve around new techniques for relating the performance of combinatorial algorithms to geometric quantities and then using a careful volumetric analysis to bound these quantities. To do so, we introduce a variety of tools from pure mathematics that are not typically used in a computer science context, including Riemann surface theory, differential and algebraic geometry, circle packing theory, geometric probability theory, harmonic analysis, and convex geometry.

The techniques advanced herein appear to be quite widely applicable and have already been applied in a variety of broader contexts [37, 43]. In one noteworthy such application, Kelner and Nikolova use random matrix theory to generalize the analysis of the simplex method to provide the first smoothed polynomial-time algorithm for a broad class on *nonconvex* optimization problems [37], providing an illustration of the wide-ranging usefulness of the ideas that we shall present.

[1] Djidjev claimed in a brief note to have such an algorithm [18], but it has never appeared in the literature.

# Part I

# A Randomized Polynomial-Time Simplex Algorithm for Linear Programming

# Chapter 2

# Introduction to Linear Programming Geometry

In this section, we shall briefly review some basic facts about linear programming geometry and the simplex method. As this material is quite standard, we shall often omit the proofs and aim only for intuition. For a more thorough treatment of the classical theory of linear programming, see Chvatal's book [13], or see Vanderbei's book [50] for a more modern viewpoint.

## 2.1 Linear Programs as Polytopes

Suppose that we are given a linear program of the form described in equation (1.1):

$$\begin{aligned} \textbf{maximize} \quad & \boldsymbol{c} \cdot \boldsymbol{x} \\ \textbf{subject to} \quad & A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{R}^d, \end{aligned}$$

where $\boldsymbol{c} \in \mathbb{R}^d$ and $\boldsymbol{b} \in \mathbb{R}^n$ are column vectors and $A$ is an $n \times d$ matrix, and let

$$P = \{\boldsymbol{x} \in \mathbb{R}^d \mid A\boldsymbol{x} \leq b\}$$

be its feasible region. Suppose further that the feasible region is nonempty and has nonempty interior. In particular, this implies that $P$ is full-dimensional and therefore is not contained in any proper linear subspace of $\mathbb{R}^d$. [1] If $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ are the rows of

[1] We make these assumptions solely to facilitate the exposition in this section; our actual algorithm will work for fully general linear programs.

$A$, then we can rewrite the feasible region as

$$P = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{a}_i \cdot x \leq b_i, \forall i\} = \bigcap_{i=1}^{n} H_i,$$

where

$$H_i = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{a}_i \cdot x \leq b_i, \forall i\}.$$

We now note a sequence of simple facts that will provide us with a dictionary to translate the given algebraic formulation of our linear program into a geometric one:

- Each of the $H_i$ is a half-space, and their intersection $P$ is a convex polyhedron[2].
- Each facet of $P$ may be given as the set of points at which some constraint $\{\boldsymbol{a}_i \cdot \boldsymbol{x} \leq b_i\}$ is tight (*i.e.*, where it is satisfied with equality).
- Each face of codimension $k$ may be given as the set of points at which some collection of $k$ constraints is tight. In particular, every vertex is the point at which a collection of $d$ constaints is tight.
- If $P$ is bounded, the objective function will have a finite maximum.
- Since the objective function is linear and $P$ is convex, every local maximum is a global maximum as well.
- If the objective function has a finite maximum, all points at which it is maximized occur on the boundary of $P$. The collection of such points constitutes a proper face of $P$ and, in particular, contains some vertex of $P$.

*Remark* 2.1.1. The last fact implies that it suffices to search over the vertices of $P$ to find the optimum. We shall make significant use of this in Chapter 3 when we introduce the simplex method.

## 2.2 Duality

In this section, we shall introduce one of the basic tools for linear programming: duality. Given any linear program, linear programming duality allows us to construct a second linear program (in a slightly modified form) that appears quite different from the original but actually has the same optimal value.

[2]From here on, all polyhedra shall be assumed convex, unless otherwise noted.

**Definition 2.2.1.** Let $\mathcal{P}$ be the linear program

$$\begin{aligned} &\textbf{maximize} && \boldsymbol{c} \cdot \boldsymbol{x} \\ &\textbf{subject to} && A\boldsymbol{x} \leq \boldsymbol{b}, \ \boldsymbol{x} \in \mathbb{R}^d \end{aligned}$$

Its *dual* is the linear program $\mathcal{D}$ given by

$$\begin{aligned} &\textbf{minimize} && \boldsymbol{b} \cdot \boldsymbol{y} \\ &\textbf{subject to} && A^T\boldsymbol{y} = \boldsymbol{c}, \ \boldsymbol{y} \geq \boldsymbol{0}. \end{aligned}$$

We shall call the original linear program the *primal* linear program when we wish to contrast it with the dual.

**Theorem 2.2.2** (Strong Linear Programming Duality). *The primal and dual linear programs have the same optimal value. That is, if either the primal or dual program is feasible with a finite optimum, then both are feasible with finite optima. In this case, if $\boldsymbol{x}_0$ is the point that maximizes $\boldsymbol{c} \cdot \boldsymbol{x}$ in the primal program and $\boldsymbol{y}_0$ is the point that minimizes $\boldsymbol{b} \cdot \boldsymbol{y}$ in the dual program, then $\boldsymbol{c} \cdot \boldsymbol{x}_0 = \boldsymbol{b} \cdot \boldsymbol{y}_0$.*

**Lemma 2.2.3** (Weak Linear Programming Duality). *For any $\boldsymbol{x}_0$ that is feasible for $\mathcal{P}$ and any $\boldsymbol{y}_0$ that is feasible for $\mathcal{D}$,*

$$\boldsymbol{c} \cdot \boldsymbol{x}_0 \leq \boldsymbol{b} \cdot \boldsymbol{y}_0.$$

*In particular, this inequality holds for $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ as described in the statement of Theorem 2.2.2.*

*Proof of Lemma 2.2.3.* Since $\boldsymbol{x}_0$ is feasible for $\mathcal{P}$, we have that $A\boldsymbol{x}_0 \leq \boldsymbol{b}$. Any $\boldsymbol{y}_0$ that is feasible for $\mathcal{D}$ has all positive components, so multiplying this inequality on the left by $\boldsymbol{y}_0^T$ yields

$$\boldsymbol{y}_0^T A\boldsymbol{x}_0 \leq \boldsymbol{y}_0^T \boldsymbol{b}. \tag{2.1}$$

However, the feasibility of $\boldsymbol{y}_0$ implies that $\boldsymbol{y}_0^T A = \boldsymbol{c}^T$. Combining this with equation 2.1 yields

$$\boldsymbol{c}^T \boldsymbol{x}_0 = \boldsymbol{y}_0^T A\boldsymbol{x}_0 \leq \boldsymbol{y}_0^T \boldsymbol{b},$$

as desired. □

We now use weak duality to sketch a proof of strong duality. The argument used here is a slightly nonstandard one; it is drawn from Schrijver's book on linear and integer programming [45].

*Sketch of Proof of Theorem 2.2.2.* Let the polyhedron $P$ be the feasible region of $\mathcal{P}$. We assume here that $P$ is nonempty and bounded. The other cases inhere no significant additional difficulties and are omitted for concision.

Our proof sketch is based on Newtonian physics[3]. We place a ball inside of our polyhedron $P$, and we subject this ball to a "gravitational" force with the same magnitude and direction as $\boldsymbol{c}$. We then let the ball roll down to its resting place, which will be the point $\boldsymbol{x}_0$ that maximizes the dot product $\boldsymbol{c} \cdot \boldsymbol{x}_0$, and we analyze the forces on the ball at equilibrium.

In order for the ball to be at rest, the total force on the ball must equal zero. Facets may only exert forces in their normal directions, and the forces may only be directed inward. As such, the $i^{\text{th}}$ facet may only exert a force $\boldsymbol{f}_i$ in the direction of $\boldsymbol{a}_i$, and this force must have a negative dot product with $\boldsymbol{a}_i$. We can thus define a vector $\boldsymbol{y}_0$ with all positive components such that $\boldsymbol{f}_i = -y_{0i}\boldsymbol{a}_i$ for all $i$.

Since the total force on the ball must equal zero, we have

$$\boldsymbol{0}^T = \boldsymbol{c}^T + \sum_i \boldsymbol{f}_i = \boldsymbol{c}^T - \sum_i y_{0i}\boldsymbol{a}_i = \boldsymbol{c}^T - \boldsymbol{y}_0^T A,$$

and thus

$$A^T \boldsymbol{y}_0 = \boldsymbol{c}.$$

It therefore follows that $\boldsymbol{y}_0$ is feasible for $\mathcal{D}$.

Now, the only facets that can exert a nonzero force on the ball are the ones that are touching it, *i.e.*, those $i$ for which $\boldsymbol{a}_i \cdot \boldsymbol{x}_0 = b_i$. This is equivalent to the statement that

$$(\boldsymbol{a}_i \cdot \boldsymbol{x}_0 - b_i)y_i = 0 \quad \text{for all } i,$$

or, written in matrix form,

$$\boldsymbol{y}_0^T(A\boldsymbol{x}_0 - \boldsymbol{b}) = \boldsymbol{0}.$$

It thus follows that we have a point $\boldsymbol{x}_0$ that is feasible for $\mathcal{P}$ and a point $\boldsymbol{y}_0$ that is feasible for $\mathcal{D}$ for which

$$\boldsymbol{c}^T\boldsymbol{x}_0 = \boldsymbol{y}_0^T A\boldsymbol{x}_0 = \boldsymbol{y}_0^T\boldsymbol{b}.$$

This implies that the maximum value of $\boldsymbol{c} \cdot \boldsymbol{x}$ in $\mathcal{P}$ is greater than or equal to the minimum value of $\boldsymbol{b} \cdot \boldsymbol{y}$ in $\mathcal{D}$. Weak duality implies the opposite inequality, and the desired theorem follows. □

*Remark* 2.2.4. By weak linear programming duality, every feasible point of the dual

[3]This recourse to physics is not fully rigorous and leaves us with something between an intuition and a proof. Nevertheless, our intuition may easily be translated into a rigorous argument; see Schrijver's book [45] for the details.

program yields a finite upper bound on the maximum of the primal program, and every feasible point of the primal program yields a finite lower bound on the minimum of the dual program. Furthermore, the argument used in the proof of strong duality shows that a finite optimum for one program yields a feasible point for the other. It thus follows that $\mathcal{P}$ is bounded if and only if $\mathcal{D}$ is feasible, and $\mathcal{D}$ is bounded if and only if $\mathcal{P}$ is feasible.

## 2.3 Polarity

In this section, we shall consider another type of duality operation known as polarity, which operates on convex polyhedra, not linear programs. While it is sometimes referred to as polyhedron duality, we stress that polarity bears no relation to linear programming duality and is a completely different operation.

Polarity may actually be defined on the larger class of arbitrary convex bodies, but for our purposes it will suffice to restrict our attention to polyhedra containing the origin in their interiors. Any such polyhedron can be described as $\{x \in \mathbb{R}^d \mid \boldsymbol{a}_i \cdot \boldsymbol{x} \leq b_i, i = 1, \ldots, n\}$ where *all of the $b_i$ are strictly positive* and the $\boldsymbol{a}_i$ span $\mathbb{R}^d$. For an exposition of the general theory, see the book by Bonneson and Fenchel /citeBF.

**Definition 2.3.1.** Let $P = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{a}_i \cdot \boldsymbol{x} \leq b_i, i = 1, \ldots, n\}$ be a polyhedron with $b_i > 0$ for all $i$. Its *polar* $P^*$ is the polyhedron given by the convex hull

$$P^* = \text{conv}(\boldsymbol{a}_1/b_1, \ldots, \boldsymbol{a}_n/b_n).$$

## 2.4 When is a Linear Program Unbounded?

Suppose that we are given a linear program with feasible region $P = \{\boldsymbol{x} \in \mathbb{R}^d \mid A\boldsymbol{x} \leq \boldsymbol{b}\}$ with $\boldsymbol{b} > 0$. In this section, we take up the question of when $P$ is unbounded. It turns out that there is a very simple criterion for this in terms of the polar polytope $P^*$.

**Theorem 2.4.1.** *Let $P$ be as above. $P$ is unbounded if and only if there exists a vector $\boldsymbol{q} \in \mathbb{R}^d$ such that the polar polytope $P^*$ is contained in the halfspace $H_{\boldsymbol{q}} = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{q} \cdot \boldsymbol{x} \leq 0\}$.*

*Proof.* Suppose first that $P$ is unbounded. Since $P$ contains the origin and is convex, this implies that there exists some vector $\boldsymbol{q}$ for which the ray $\boldsymbol{r}_{\boldsymbol{q}} := \{t\boldsymbol{q} \mid t \geq 0\}$ extends off to infinity while remaining inside of $P$. We claim that this implies that $P^* \subseteq H_{\boldsymbol{q}}$.

To see this, suppose to the contrary that $P^* \not\subseteq H_{\boldsymbol{q}}$. This implies that there exists some $\boldsymbol{a}_i$ for which $\boldsymbol{a}_i/b_i \notin H_{\boldsymbol{q}}$, *i.e.*, for which $(\boldsymbol{a}_i/b_i) \cdot \boldsymbol{q} > 0$. In this case, the corresponding inequality $\boldsymbol{a}_i \cdot \boldsymbol{x} \leq b_i$ will be violated by the point $t\boldsymbol{q}$ whenever $t > b_i/(\boldsymbol{a}_i \cdot \boldsymbol{q})$, contradicting the infinitude of the ray $\boldsymbol{r}_{\boldsymbol{q}}$.

For the converse, we shall suppose that $P$ is bounded and shall deduce that $P^*$ is not contained in any halfspace through the origin. Indeed, this follows from the same argument as above: if $P^*$ were contained in the halfspace $H_{\boldsymbol{q}}$ then the ray $\boldsymbol{r}_{\boldsymbol{q}}$ would extend off to infinity, which would contradict the presumed boundedness of $P$. □

A convex body is contained in a half-space if and only if it does not contain the origin in its interior. We thus deduce:

**Corollary 2.4.2.** *$P$ is bounded if and only if $P^*$ contains the origin in its interior.*

*Remark* 2.4.3. Corollary 2.4.2 allows us to produce a *certificate of boundedness* for $P$ by expressing the origin as a convex combination of the $\boldsymbol{a}_i$ *with all strictly positive coefficients.* (The positivity of the coefficients guarantees that the origin is contained in the interior and not on the boundary of $P^*$. Here, we use our assumption that the $\boldsymbol{a}_i$ span $\mathbb{R}^d$.) We shall make use of this in Chapters 5 and 6.

# Chapter 3

# The Simplex Algorithm

In this chapter, we shall introduce our primary object of study, the simplex algorithm. As this material is standard and widely available, we shall restrict our discussion to a high-level overview. For an implementation-level discussion of the simplex method, we refer the reader to Vanderbei's book [50].

## 3.1 The General Method

As we saw in Section 2.1, the feasible region of a linear program is a polytope $P$, the objective function achieves its maximum at a vertex of $P$, and the objective function has no nonglobal local maxima. It thus suffices to search among the vertices for a local maximum of the objective function.

Since $P$ has finitely many vertices, this suggests an obvious algorithm for linear programming, known as the *simplex algorithm* or *simplex method*. Simply neglect all of the higher-dimensional faces of $P$ and just consider the graph of vertices and edges of $P$. Start at some vertex and walk along the edges of the graph until you find a vertex that is a local (and thus global) maximum.

Of course, the above is really just a meta-algorithm. To make it into a fully specified algorithm, one must further specify two things:

1. How does one obtain the starting vertex?

2. Given the current vertex, how does one obtain the next vertex? This is known as the *pivot rule*.

Since the simplex method was first introduced, this definition has been broadened to allow the algorithm to walk on other graphs associated with the polytope; noteworthy examples of such algorithms include the self-dual simplex method and the criss-cross

method [15, 22]. Nevertheless, while there have been numerous pivot rules set forth for linear programming, up until the present work none have been shown to terminate in a polynomial number of steps.

In the following section, we shall describe a classical pivot rule known as the "shadow-vertex method." We stress that this pivot rule does *not* always terminate in a polynomial number of steps. Instead, we shall use it as a component of a more complicated algorithm that does indeed have the desired polynomial running time.

## 3.2 The Shadow-Vertex Method

Let $P$ be a convex polyhedron, and let $S$ be a two-dimensional subspace. The *shadow* of $P$ onto $S$ is simply the projection of $P$ onto $S$. The shadow is a polygon, and every vertex (edge) of the polygon is the image of some vertex (edge) of $P$. One can show that the set of vertices of $P$ that project onto the boundary of the shadow polygon are exactly the vertices of $P$ that optimize objective functions in $S$ [6, 24].

These observations are the inspiration for the shadow-vertex simplex method, which lifts the simplicity of linear programming in two dimensions to the general case [6, 24]. To start, the shadow-vertex method requires as input a vertex $\boldsymbol{v}_0$ of $P$. It then chooses some objective function optimized at $\boldsymbol{v}_0$, say $\boldsymbol{f}$, sets $S = \text{span}(\boldsymbol{c}, \boldsymbol{f})$, and considers the shadow of $P$ onto $S$. If no degeneracies occur, then for each vertex $\boldsymbol{y}$ of $P$ that projects onto the boundary of the shadow, there is a unique neighbor of $\boldsymbol{y}$ on $P$ that projects onto the next vertex of the shadow in clockwise-order. Thus, by tracing the vertices of $P$ that map to the boundary of the shadow, the shadow-vertex method can move from the vertex it knows that optimizes $\boldsymbol{f}$ to the vertex that optimizes $\boldsymbol{c}$. The number of steps that the method takes will be bounded by the number of edges of the shadow polygon. For future reference, we call the shadow-vertex simplex method by

$$\textsc{ShadowVertex}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n, \boldsymbol{b}, \boldsymbol{c}, S, \boldsymbol{v}_0, s),$$

where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, $\boldsymbol{b}$, and $\boldsymbol{c}$ specify a linear program of form (1.1), $S$ is a two-dimensional subspace containing $\boldsymbol{c}$, and $\boldsymbol{v}_0$ is the start vertex, which must optimize some objective function in $S$. We allow the method to run for at most $s$ steps. If it has not found the vertex optimizing $\boldsymbol{c}$ within that time, it should return $(\textbf{fail}, \boldsymbol{y})$, where $\boldsymbol{y}$ is its current vertex. If it has solved the linear program, it either returns $(\textbf{opt}, \boldsymbol{x})$, where $\boldsymbol{x}$ is the solution, or **unbounded** if it was unbounded.

# Chapter 4

# Bounding the Shadow Size

In this chapter, we shall show that if the polytope is in a "good" coordinate system and the distances of the facets from the origin are randomly perturbed, then the number of edges of the shadow onto a random subspace $S$ is expected to be polynomial. We shall then provide a slight generalization of this theorem that we will need in the analysis of our algorithm. The one geometric fact that we will require in our analysis is that if an edge of $P$ is tight for inequalities $\boldsymbol{a}_i \cdot \boldsymbol{x} = b_i$, for $i \in I$, then the edge projects to an edge in the shadow if and only if $S$ intersects the convex hull of $\{\boldsymbol{a}_i\}_{i \in I}$. Below, we shall often abuse notation by identifying an edge with the set of constraints $I$ for which it is tight.

## 4.1 The Shadow Size in the $k$-Round Case

**Definition 4.1.1.** We say that a polytope $P$ is $k$-round if

$$B(\boldsymbol{0}, 1) \subseteq P \subseteq B(\boldsymbol{0}, k),$$

where $B(\boldsymbol{0}, r)$ is the ball of radius $r$ centered at the origin.

In this section, we will consider a polytope $P$ defined by

$$\left\{\boldsymbol{x} | \forall i,\ \boldsymbol{a}_i^T \boldsymbol{x} \leq 1\right\},$$

in the case that $P$ is $k$-round. Note that the condition $B(\boldsymbol{0}, 1) \subseteq P$ implies $\|\boldsymbol{a}_i\| \leq 1$.

We will then consider the polytope we get by perturbing the right-hand sides,

$$Q = \left\{\boldsymbol{x} | \forall i,\ \boldsymbol{a}_i^T \boldsymbol{x} \leq 1 + r_i\right\},$$

where each $r_i$ is an independent exponentially distributed random variable with ex-

pectation $\lambda$. That is,

$$\Pr[r_i \geq t] = e^{-t/\lambda}$$

for all $t \geq 0$.

Note that we will eventually set $\lambda$ to $1/n$, but could obtain stronger bounds by setting $\lambda = c \log n$ for some constant $c$.

We will prove that the expected number of edges of the projection of $Q$ onto a random 2-plane is polynomial in $n$, $k$ and $1/\lambda$. In particular, this will imply that for a random objective function, the shortest path from the minimum vertex to the maximum vertex is expected to have a number of steps polynomial in $n$, $k$ and $1/\lambda$.

Our proof will proceed by analyzing the expected length of edges that appear on the boundary of the projection. We shall show that the total length of all such edges is expected to be bounded above. However, we shall also show that our perturbation will cause the expected length of each edge to be reasonably large. Combining these two statements will provide a bound on the expected number of edges that appear.

**Theorem 4.1.2.** *Let $\boldsymbol{v}$ and $\boldsymbol{w}$ be uniformly random unit vectors, and let $V$ be their span. Then, the expectation over $\boldsymbol{v}$, $\boldsymbol{w}$, and the $r_i$s of the number of facets of the projection of $Q$ onto $V$ is at most*

$$\frac{12\pi k(1 + \lambda \ln(ne))\sqrt{d}n}{\lambda}.$$

*Proof.* We first observe that the perimeter of the shadow of $P$ onto $V$ is at most $2\pi k$. Let $r = \max_i r_i$. Then, as

$$Q \subseteq \left\{\boldsymbol{x} | \forall i,\ \boldsymbol{a}_i^T \boldsymbol{x} \leq 1 + r\right\} = (1 + r)P,$$

the perimeter of the shadow of $Q$ onto $V$ is at most $2\pi k(1 + r)$. As we shall show in Proposition 4.1.3, the expectation of $r$ is at most $\lambda \ln(ne)$, so the expected perimeter of the shadow of $Q$ on $V$ is at most $2\pi k(1 + \lambda \ln(ne))$.

Now, each edge of $Q$ is determined by the subset of $d - 1$ of the constraints that are tight on that edge. For each $I \in \binom{[n]}{d-1}$, let $S_I(V)$ be the event that edge $I$ appears in the shadow, and let $\ell(I)$ denote the length of that edge in the shadow. We now know

$$\begin{aligned} 2\pi k(1 + \lambda \ln(ne)) &\geq \sum_{I \in \binom{[n]}{d-1}} \mathbf{E}[\ell(I)] \\ &= \sum_{I \in \binom{[n]}{d-1}} \mathbf{E}[\ell(I) | S_I(V)] \Pr[S_I(V)]. \end{aligned}$$

Below, in Lemma 4.1.9, we will prove that

$$\mathbf{E}\left[\ell(I)|S_I(V)\right] \geq \frac{\lambda}{6\sqrt{dn}}.$$

From this, we conclude that

$$\begin{aligned}\mathbf{E}\left[\text{number of edges}\right] &= \sum_{I \in \binom{[n]}{d-1}} \Pr\left[S_I(V)\right] \\ &\leq \frac{12\pi k(1+\lambda \ln(ne))\sqrt{dn}}{\lambda},\end{aligned}$$

as desired. □

We now prove the various lemmas used in the proof of Theorem 4.1.2. Our first is a straightforward statement about exponential random variables.

**Proposition 4.1.3.** *Let $r_1, \ldots, r_n$ be independent exponentially distributed random variables of expectation $\lambda$. Then,*

$$\mathbf{E}\left[\max r_i\right] \leq \lambda \ln(ne).$$

*Proof.* This follows by a simple calculation, in which the first inequality follows from a union bound:

$$\begin{aligned}\mathbf{E}\left[\max r_i\right] &= \int_{t=0}^{\infty} \Pr\left[\max r_i \geq t\right] \\ &\leq \int_{t=0}^{\infty} \Pr\left[\min(1, ne^{-t/\lambda})\right] \\ &= \int_{t=0}^{\lambda \ln n} 1 + \int_{\lambda \ln n}^{\infty} ne^{-t/\lambda} \\ &= (\lambda \ln n) + \lambda \\ &= \lambda \ln(ne),\end{aligned}$$

as desired. □

We shall now prove the lemmas necessary for Lemma 4.1.9, which bounds the expected length of an edge, given that it appears in the shadow. Our proof of Lemma 4.1.9 will have two parts. In Lemma 4.1.7, we will show that it is unlikely that the edge indexed by $I$ is short, given that it appears on the convex hull of $Q$. We will then use Lemma 4.1.8 to show that, given that it appears in the shadow,

it is unlikely that its projection onto the shadow plane is much shorter. To facilitate the proofs of these lemmas, we shall prove some auxiliary lemmas about *shifted exponential random variables.*

**Definition 4.1.4.** We say that $r$ is a shifted exponential random variable with parameter $\lambda$ if there exists a $t \in \mathbb{R}$ such that $r = s - t$, where $s$ is an exponential random variable with expectation $\lambda$.

**Proposition 4.1.5.** *Let $r$ be a shifted exponential random variable of parameter $\lambda$. Then, for all $q \in \mathbb{R}$ and $\epsilon \geq 0$,*

$$\Pr\left[r \leq q + \epsilon \middle| r \geq q\right] \leq \epsilon/\lambda.$$

*Proof.* As $r - q$ is a shifted exponential random variable, it suffices to consider the case in which $q = 0$. So, assume $q = 0$ and $r = s - t$, where $s$ is an exponential random variable of expectation $\lambda$. We now need to compute

$$\Pr\left[s \leq t + \epsilon \middle| s \geq t\right]. \tag{4.1}$$

We only need to consider the case $\epsilon < \lambda$, as the proposition is trivially true otherwise. We first consider the case in which $t \geq 0$. In this case, we have

$$\begin{aligned}(4.1) = \Pr\left[s \leq t + \epsilon \middle| s \geq t\right] &= \frac{\frac{1}{\lambda}\int_{s=t}^{t+\epsilon} e^{-s/\lambda} ds}{\frac{1}{\lambda}\int_{s=t}^{\infty} e^{-s/\lambda} ds} \\ &= \frac{e^{-t/\lambda} - e^{-t/\lambda - \epsilon/\lambda}}{e^{-t/\lambda}} \\ &= 1 - e^{\epsilon/\lambda} \leq \epsilon/\lambda,\end{aligned}$$

for $\epsilon/\lambda \leq 1$.

Finally, the case when $t \leq 0$ follows from the analysis in the case $t = 0$. □

**Lemma 4.1.6.** *For $N$ and $P$ disjoint subsets of $\{1, \ldots, n\}$, let $\{r_i\}_{i \in P}$ and $\{r_j\}_{j \in N}$ be independent random variables, each of which is a shifted exponential random variable with parameter at least $\lambda$. Then*

$$\Pr\left[\min_{i \in P}(r_i) + \min_{j \in N}(r_j) < \epsilon \middle| \min_{i \in P}(r_i) + \min_{j \in N}(r_j) \geq 0\right] \leq n\epsilon/2\lambda.$$

*Proof.* Assume without loss of generality that $|P| \leq |N|$, so $|P| \leq n/2$.

Set $r^+ = \min_{i \in P} r_i$ and $r^- = \min_{j \in N} r_j$. Sample $r^-$ according to the distribution induced by the requirement that $r^+ + r^- \geq 0$. Given the sampled value for $r^-$, the induced distribution on $r^+$ is simply the base distribution restricted to the space where $r^+ \geq -r^-$. So, it suffices to bound

$$\begin{aligned}
&\max_{r^-} \Pr_{r^+} \left[ r^+ < \epsilon - r^- \middle| r^+ \geq -r^- \right] \\
&= \max_{r^-} \Pr_{r_i : i \in P} \left[ \min_{i \in P}(r_i) < \epsilon - r^- \middle| \min_{i \in P}(r_i) \geq -r^- \right] \\
&\leq \max_{r^-} \sum_{k \in P} \Pr_{r_i : i \in P} \left[ r_k < \epsilon - r^- \middle| \min_{i \in P}(r_i) \geq -r^- \right] \\
&= \sum_{k \in P} \max_{r^-} \Pr_{r_i : i \in P} \left[ r_k < \epsilon - r^- \middle| \min_{i \in P}(r_i) \geq -r^- \right] \\
&= \sum_{k \in P} \max_{r^-} \Pr_{r_k} \left[ r_k < \epsilon - r^- \middle| r_k \geq -r^- \right] \\
&\leq |P| \, (\epsilon / \lambda),
\end{aligned}$$

where the last equality follows from the independence of the $r_i$'s, and the last inequality follows from Proposition 4.1.5. □

**Lemma 4.1.7.** *Let $I \in \binom{[n]}{d-1}$, and let $A(I)$ be the event that $I$ appears on the convex hull of $Q$. Let $\delta(I)$ denote the length of the edge $I$ on $Q$. Then,*

$$\Pr \left[ \delta(I) < \epsilon | A(I) \right] \leq \frac{n\epsilon}{2\lambda}.$$

*Proof.* Without loss of generality, we set $I = \{1, \ldots, d-1\}$. As our proof will not depend upon the values of $r_1, \ldots, r_{d-1}$, assume that they have been set arbitrarily. Now, parameterize the line of points satisfying

$$\boldsymbol{a}_i^T \boldsymbol{x} = 1 + r_i, \quad \text{for } i \in I,$$

by

$$l(t) := \boldsymbol{p} + t\boldsymbol{q},$$

where $\boldsymbol{p}$ is the point on the line closest to the origin, and $\boldsymbol{q}$ is a unit vector orthogonal to $\boldsymbol{p}$. For each $i \geq d$, let $t_i$ index the point where the $i^{\text{th}}$ constraint intersects the line, *i.e.*,

$$\boldsymbol{a}_i^T l(t_i) = 1 + r_i. \tag{4.2}$$

Now, divide the constraints indexed by $i \notin I$ into a positive set, $P = \left\{ i \geq d | \boldsymbol{a}_i^T \boldsymbol{q} \geq 0 \right\}$, and a negative set $N = \left\{ i \geq d | \boldsymbol{a}_i^T \boldsymbol{q} < 0 \right\}$. Note that each constraint in the positive

set is satisfied by $l(-\infty)$ and each constraint in the negative set is satisfied by $l(\infty)$. The edge $I$ appears in the convex hull if and only if for each $i \in P$ and $j \in N$, $t_j < t_i$. When the edge $I$ appears, its length is

$$\min_{i \in P, j \in N} t_i - t_j.$$

Solving (4.2) for $i \in P$, we find $t_i = \frac{1}{\boldsymbol{a}_i^T \boldsymbol{q}} \left(1 - \boldsymbol{a}_i^T \boldsymbol{p} + r_i\right)$. Similarly, for $j \in N$, we find $t_j = \frac{1}{|\boldsymbol{a}_j^T \boldsymbol{q}|} \left(-1 + \boldsymbol{a}_j^T \boldsymbol{p} - r_j\right)$. Thus, $t_i$ for $i \in P$ and $-t_j$ for $j \in N$ are both shifted exponential random variables with parameter at least $\lambda$. So, by Lemma 4.1.6,

$$\Pr_{\{r_i | i \notin I\}} \left[ \min_{i \in P, j \in N} t_i - t_j < \epsilon | A(I) \right] < n\epsilon/2\lambda.$$

□

**Lemma 4.1.8.** *Let $Q$ be an arbitrary polytope, and let $I$ index an edge of $Q$. Let $\boldsymbol{v}$ and $\boldsymbol{w}$ be random unit vectors, and let $V$ be their span. Let $S_I(V)$ be the event that the edge $I$ appears on the convex hull of the projection of $Q$ onto $V$. Let $\theta_I(V)$ denote the angle of the edge $I$ to $V$. Then*

$$\Pr_{\boldsymbol{v}, \boldsymbol{w}} [\cos(\theta_I(V)) < \epsilon | S_I(V)] \leq d\epsilon^2.$$

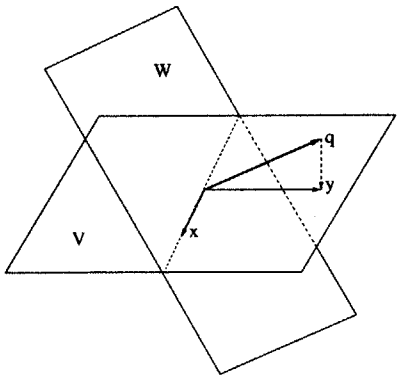


Figure 4-1: The points $x$, $y$ and $q$.

*Proof.* As in the proof of Lemma 4.1.7, parameterize the edge by

$$l(t) := \boldsymbol{p} + t\boldsymbol{q},$$

where $\boldsymbol{q}$ is a unit vector. Observe that $S_I(V)$ holds if and only if $V$ non-trivially intersects the cone $\left\{\sum_{i\in I} \alpha_i \boldsymbol{a}_i | \alpha_i \geq 0\right\}$, which we denote $C$. To evaluate the probability, we will perform a change of variables that will both enable us to easily evaluate the angle between $\boldsymbol{q}$ and $V$ and to determine whether $S_I(V)$ holds. Some of the new variables that we introduce are shown in Figure 4-1.

First, let $W$ be the span of $\{\boldsymbol{a}_i | i \in I\}$, and note that $W$ is also the subspace orthogonal to $\boldsymbol{q}$. The angle of $\boldsymbol{q}$ to $V$ is determined by the angle of $\boldsymbol{q}$ to the unit vector through the projection of $\boldsymbol{q}$ onto $V$, which we will call $\boldsymbol{y}$. Fix any vector $\boldsymbol{c} \in C$, and let $\boldsymbol{x}$ be the unique unit vector in $V$ that is orthogonal to $\boldsymbol{y}$ and has positive inner product with $\boldsymbol{c}$. Note that $\boldsymbol{x}$ is also orthogonal to $\boldsymbol{q}$, and so $\boldsymbol{x} \in V \cap W$. Also note that $S_I(V)$ holds if and only if $\boldsymbol{x} \in C$.

Instead of expressing $V$ as the span of $\boldsymbol{v}$ and $\boldsymbol{w}$, we will express it as the span of $\boldsymbol{x}$ and $\boldsymbol{y}$, which are much more useful vectors. In particular, we need to express $\boldsymbol{v}$ and $\boldsymbol{w}$ in terms of $\boldsymbol{x}$ and $\boldsymbol{y}$, which we do by introducing two more variables, $\alpha$ and $\beta$, so that

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{x}\cos\alpha + \boldsymbol{y}\sin\alpha, \text{and} \\ \boldsymbol{w} &= \boldsymbol{x}\cos\beta + \boldsymbol{y}\sin\beta. \end{aligned}$$

Note that number of degrees of freedom has not changed: $\boldsymbol{v}$ and $\boldsymbol{w}$ each had $d-1$ degrees of freedom, while $\boldsymbol{x}$ only has $d-2$ degrees of freedom since it is restricted to be orthogonal to $\boldsymbol{q}$, and given $\boldsymbol{x}$, $\boldsymbol{y}$ only has $d-2$ degrees of freedom since it is restricted to be orthogonal to $\boldsymbol{x}$.

We now make one more change of variables so that the angle between $\boldsymbol{q}$ and $\boldsymbol{y}$ becomes a variable. To do this, we let $\theta = \theta_I(V)$ be the angle between $\boldsymbol{y}$ and $\boldsymbol{q}$, and note that once $\theta$ and $\boldsymbol{x}$ have been specified, $\boldsymbol{y}$ is constrained to lie on a $d-2$ dimensional sphere. We let $\boldsymbol{z}$ denote the particular point on that sphere.

Deshpande and Spielman [16, Full version] prove that the Jacobian of this change of variables from $\boldsymbol{v}$ and $\boldsymbol{w}$ to $\alpha$, $\beta$, $\boldsymbol{x}$, $\theta$, $\boldsymbol{z}$ is

$$c(\cos\theta)(\sin\theta)^{d-3}\sin(\alpha-\beta)^{d-2},$$

where $c$ is a constant depending only on the dimension.

We now compute

$$\begin{aligned}
&\Pr_V[\cos(\theta_I(V)) < \epsilon | S_I(V)] \\
&= \frac{\int_{\boldsymbol{v},\boldsymbol{w}\in S^{n-1}:V\cap C\neq\emptyset \text{ and } \theta_I(V)\leq\epsilon} 1\, d\boldsymbol{v}\; d\boldsymbol{w}}{\int_{\boldsymbol{v},\boldsymbol{w}\in S^{n-1}:\mathrm{Span}(\boldsymbol{v},\boldsymbol{w})\cap C\neq\emptyset} 1\, d\boldsymbol{v}\; d\boldsymbol{w}} \\
&= \frac{\int_{\substack{\theta>\cos^{-1}(\epsilon),\\ \boldsymbol{x}\in C,\boldsymbol{z},\alpha,\beta}} c(\cos\theta)(\sin\theta)^{d-3}\sin^{d-2}(\alpha-\beta)\, d\boldsymbol{x}\; d\boldsymbol{z}\; d\alpha\; d\beta\; d\theta}{\int_{\boldsymbol{x}\in C,\theta,\boldsymbol{z},\alpha,\beta} c(\cos\theta)(\sin\theta)^{d-3}\sin^{d-2}(\alpha-\beta)\, d\boldsymbol{x}\; d\boldsymbol{z}\; d\alpha\; d\beta\; d\theta} \\
&= \frac{\int_{\theta=\cos^{-1}(\epsilon)}^{\pi/2}(\cos\theta)(\sin\theta)^{d-3}\, d\theta}{\int_{\theta=0}^{\pi/2}(\cos\theta)(\sin\theta)^{d-3}\, d\theta} \\
&= \frac{(\sin\theta)^{d-2}\big|_{\cos^{-1}(\epsilon)}^{\pi/2}}{(\sin\theta)^{d-2}\big|_{0}^{\pi/2}} \\
&\leq 1 - (\sin(\cos^{-1}(\epsilon))^{d-2} \\
&\leq 1 - (1-\epsilon^2)^{(d-2)/2} \;\leq\; \frac{d-2}{2}\epsilon^2.
\end{aligned}$$

□

**Lemma 4.1.9.** *For all* $I \in \binom{[n]}{d-1}$,

$$\mathbf{E}_{V,r_1,\ldots,r_n}[\ell(I)|S_I(V)] \geq \frac{\lambda}{6\sqrt{d}n}.$$

*Proof.* For each edge $I$, $\ell(I) = \delta(I)\cos(\theta_I(V))$. Lemma 4.1.7 now implies that,

$$\Pr\left[\delta(I) \geq \frac{\lambda}{n} \;\middle|\; A(I)\right] \geq 1/2.$$

By Lemma 4.1.8,

$$\Pr_V\left[\cos(\theta_I(V)) \geq 1/\sqrt{2d} \;\middle|\; S_I(V)\right] \geq 1/2.$$

Given that edge $I$ appears on the shadow, it follows that $\ell(I) > (1/\sqrt{2d})\left(\frac{\lambda}{n}\right)$ with probability at least $1/4$. Thus, its expected length when it appears is at least $\frac{\lambda}{6\sqrt{d}n}$. □

## 4.2 The Shadow Size in the General Case

In this section, we present an extension of Theorem 4.1.2 that we will require in the analysis of our simplex algorithm. We extend the theorem in two ways. First of

all, we examine what happens when $P$ is not $k$-round position. In this case, we just show that the shadow of the convex hull of the vertices of bounded norm probably has few edges. As such, if we take a polynomial number of steps around the shadow, we should either come back to where we started or find a vertex far from the origin. Secondly, we consider the shadow onto random planes that come close to a particular vector, rather than just onto uniformly random planes.

**Definition 4.2.1.** For a unit vector $\boldsymbol{u}$ and a $\rho > 0$, we define the $\rho$-perturbation of $\boldsymbol{u}$ to be the random unit vector $\boldsymbol{v}$ chosen by

1. choosing a $\theta \in [0, \pi]$ according to the restriction of the exponential distribution of expectation $\rho$ to the range $[0, \pi]$, and
2. setting $\boldsymbol{v}$ to be a uniformly chosen unit vector of angle $\theta$ to $\boldsymbol{u}$.

**Theorem 4.2.2.** *Let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ be vectors of norm at most 1. Let $r_1, \ldots, r_n$ be independent exponentially distributed random variables with expectation $\lambda$. Let $Q$ be the polytope given by*

$$Q = \left\{ \boldsymbol{x} | \forall i,\ \boldsymbol{a}_i^T \boldsymbol{x} \leq 1 + r_i \right\}.$$

*Let $\boldsymbol{u}$ be an arbitrary unit vector, $\rho < 1/\sqrt{d}$, and let $\boldsymbol{v}$ be a random $\rho$ perturbation of $\boldsymbol{u}$. Let $\boldsymbol{w}$ be a uniformly chosen random unit vector. Then, for all $t > 1$,*

$$\mathbf{E}_{r_1,\ldots,r_n,\boldsymbol{v},\boldsymbol{w}} \left[ \text{ShadowSize}_{\text{span}(\boldsymbol{v},\boldsymbol{w})}(Q \cap B(\boldsymbol{0}, t)) \right] \quad \leq \quad \frac{42\pi t(1 + \lambda \log n)\sqrt{d}n}{\lambda \rho}.$$

*Proof.* The proof of Theorem 4.2.2 is almost identical to that of Theorem 4.1.2, except that we substitute Lemma 4.2.3 for Lemma 4.1.7, and we substitute Lemma 4.2.4 for Lemma 4.1.8. □

**Lemma 4.2.3.** *For $I \subseteq \binom{[n]}{d-1}$ and $t > 0$,*

$$\Pr\left[ \delta(I) < \epsilon \middle| A(I) \text{ and } I \cap B(\boldsymbol{0}, t) \neq \emptyset \right] \leq \frac{n\epsilon}{2\lambda}.$$

*Proof.* The proof is identical to the proof of Lemma 4.1.7, except that in the proof of Lemma 4.1.6 we must condition upon the events that

$$r^+ \geq -\sqrt{t - \|\boldsymbol{p}\|} \quad \text{and} \quad r^- \leq \sqrt{t - \|\boldsymbol{p}\|}.$$

These conditions have no impact on any part of the proof. □

**Lemma 4.2.4.** *Let $Q$ be an arbitrary polytope, and let $I$ index an edge of $Q$. Let $\boldsymbol{u}$ be any unit vector, let $\rho < 1/\sqrt{d}$, and let $\boldsymbol{v}$ be a random $\rho$ perturbation of $\boldsymbol{u}$. Let $\boldsymbol{w}$ be a uniformly chosen random unit vector, and let $V = \text{span}(\boldsymbol{u}, \boldsymbol{v})$. Then*

$$\Pr_{\boldsymbol{v},\boldsymbol{w}}[\cos(\theta_I(V)) < \epsilon | S_I(V)] \leq 3.5\epsilon^2/\rho^2.$$

*Proof.* We perform the same change of variables as in Lemma 4.1.8.

To bound the probability that $\cos\theta < \epsilon$, we will allow the variables $\boldsymbol{x}$, $\boldsymbol{z}$, $\alpha$ and $\beta$ to be fixed arbitrarily, and just consider what happens as we vary $\theta$. To facilitate writing the resulting probability, let $\mu$ denote the density function on $\boldsymbol{v}$. If we fix $\boldsymbol{x}$, $\boldsymbol{z}$, $\alpha$ and $\beta$, then we can write $\boldsymbol{v}$ as a function of $\theta$. Moreover, as we vary $\theta$ by $\phi$, $\boldsymbol{v}$ moves through an angle of at most $\phi$. So, for all $\phi < \rho$ and $\theta$,

$$\mu(\boldsymbol{v}(\theta)) < \mu(\boldsymbol{v}(\theta + \phi))/e. \tag{4.3}$$

With this fact in mind, we compute the probability to be

$$\begin{aligned}
&\frac{\int_{\boldsymbol{v},\boldsymbol{w}\in S^{n-1}:V\cap C\neq\emptyset \text{ and } \theta_I(V)\leq\epsilon} \mu(\boldsymbol{v})\, d\boldsymbol{v}\; d\boldsymbol{w}}{\int_{\boldsymbol{v},\boldsymbol{w}\in S^{n-1}:V\cap C\neq\emptyset} \mu(\boldsymbol{v})\, d\boldsymbol{v}\; d\boldsymbol{w}} \\
&\leq \max_{\boldsymbol{x},\boldsymbol{z},\alpha,\beta} \frac{\int_{\theta=\cos^{-1}(\epsilon)}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\mu(\boldsymbol{v}(\theta))\, d\theta}{\int_{\theta=0}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\mu(\boldsymbol{v}(\theta))\, d\theta} \\
&\leq \max_{\boldsymbol{x},\boldsymbol{z},\alpha,\beta} \frac{\int_{\theta=\cos^{-1}(\epsilon)}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\mu(\theta)\, d\theta}{\int_{\theta=\pi/2-\rho}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\mu(\theta)\, d\theta} \\
&\leq \frac{e\int_{\theta=\cos^{-1}(\epsilon)}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\, d\theta}{\int_{\theta=\pi/2-\rho}^{\pi/2} (\cos\theta)(\sin\theta)^{d-3}\, d\theta}, \text{ by (4.3)} \\
&= e\frac{(\sin\theta)^{d-2}\big|_{\cos^{-1}(\epsilon)}^{\pi/2}}{(\sin\theta)^{d-2}\big|_{\pi/2-\rho}^{\pi/2}} \\
&= e\frac{1-(\sin(\cos^{-1}(\epsilon))^{d-2}}{1-(\sin(\rho)^{d-2}} \\
&\leq e\frac{1-(1-\epsilon^2)^{(d-2)/2}}{1-(1-\rho^2/2)^{d-2}} \\
&\leq e\frac{(d-2)\epsilon^2}{(d-2)4(1-1/\sqrt{e})\rho^2}, \text{ as } \rho < 1/\sqrt{d}, \\
&\leq 3.5(\epsilon/\rho)^2.
\end{aligned}$$

□

# Chapter 5

# Reduction of Linear Programming to Certifying Boundedness

## 5.1 Reduction to a Feasibility Problem

We now recall an old trick [45, p. 125] for reducing the problem of solving a linear program in form (1.1) to a different form that will be more useful for our purposes. We recall that the dual of such a linear program is given by

$$\begin{array}{ll} \textbf{minimize} & \boldsymbol{b} \cdot \boldsymbol{y} \\ \textbf{subject to} & A^T\boldsymbol{y} = \boldsymbol{c},\ \boldsymbol{y} \geq 0, \end{array} \tag{5.1}$$

and that when the programs are both feasible and bounded, they have the same solution. Thus, any feasible solution to the system of constraints

$$\begin{array}{c} A\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{x} \in \mathbb{R}^d, \\ A^T\boldsymbol{y} = \boldsymbol{c},\ \boldsymbol{y} \geq \boldsymbol{0}, \\ \boldsymbol{c} \cdot \boldsymbol{x} = \boldsymbol{b} \cdot \boldsymbol{y} \end{array} \tag{5.2}$$

provides a solution to both the linear program and its dual. Since solving the above system is trivial if $\boldsymbol{b}$ is the zero vector, we assume from here on in that $\boldsymbol{b} \neq \boldsymbol{0}$.

By introducing a new vector of variables $\boldsymbol{\delta} \in \mathbb{R}^d$, we can replace the matrix inequality $A\boldsymbol{x} \leq \boldsymbol{b}$ by an equality constraint and a nonnegativity constraint:

$$\begin{aligned} A\boldsymbol{x} + \boldsymbol{\delta} &= \boldsymbol{b} \\ \boldsymbol{\delta} &\geq \boldsymbol{0}. \end{aligned}$$

Now the variables $\delta_i$ and $y_i$ are constrained to be nonnegative, whereas each $x_i$ may be

positive or negative. We would like to convert our system so that all of our variables are constrained to be nonnegative. To do this, we replace each variable $x_i$ by a pair of variables, $x_i^+$ and $x_i^-$, each of which is constrained to be nonnegative. We then replace every occurrence of the variable $x_i$ with the difference $x_i^+ - x_i^-$. It is not difficult to see that, at any finite optimum, one of the two variables will be zero and the other will equal the magnitude of the value that $x_i$ would have assumed at the optimum of the original system.

Collecting all of our variables into one vector $\boldsymbol{z}_1$ now gives us a feasibility problem of the form

$$\begin{aligned} A_1^T \boldsymbol{z}_1 &= \boldsymbol{b}_1 \\ \boldsymbol{z}_1 \geq \boldsymbol{0}, \boldsymbol{z}_1 &\neq \boldsymbol{0} \end{aligned} \tag{5.3}$$

where $A_1$ is a matrix constructed from $A$, $\boldsymbol{b}$, and $\boldsymbol{c}$, and the vector $\boldsymbol{b}_1$ is not the zero vector. If $\boldsymbol{a}_1^{(1)}, \ldots, \boldsymbol{a}_1^{(n)}$ are the rows of $A_1$, expressed as column vectors, we can write this as

$$\begin{aligned} \textstyle\sum_{i \geq 1} z_{1,i} \boldsymbol{a}_1^{(i)} &= \boldsymbol{b}_1 \\ \boldsymbol{z}_1 \geq \boldsymbol{0}, \boldsymbol{z}_1 &\neq \boldsymbol{0}. \end{aligned} \tag{5.4}$$

**Lemma 5.1.1.** *Solving the system in* (5.4) *can be reduced in polynomial time to solving a system of the form*

$$\begin{aligned} A_2^T \boldsymbol{z}_2 &= \boldsymbol{0} \\ \boldsymbol{z}_2 \geq \boldsymbol{0}, \boldsymbol{z}_2 &\neq \boldsymbol{0}. \end{aligned} \tag{5.5}$$

*Proof.* Suppose that the bit-length required to express the system in (5.4) is $L$. It is a standard fact in the analysis of linear programs that if (5.4) has a solution then there is a value $\kappa = \kappa(L)$ that is singly-exponential in $L$ so that (5.4) has a solution with $||\boldsymbol{z}_2||_1 < \kappa$ [50]. Using this value of $\kappa$, add a new coordinate $z_{2,0}$ and form the system

$$\begin{aligned} -z_{2,0} \boldsymbol{b}_1 + \textstyle\sum_{i \geq 1} z_{2,i} \left( \boldsymbol{a}_1^{(i)} - \frac{1}{\kappa} \boldsymbol{b}_1 \right) &= \boldsymbol{0} \\ \boldsymbol{z}_2 \geq \boldsymbol{0}, \boldsymbol{z}_2 &\neq \boldsymbol{0}. \end{aligned} \tag{5.6}$$

We claim that the system in (5.6) is feasible if and only if the system in (5.4) is.

To see this, first suppose that $\boldsymbol{z}_1$ is a solution to the system in (5.4), and let

$$z_{2,i} = \begin{cases} z_{1,i} & \text{for } i \geq 1 \\ 1 - \frac{1}{\kappa}||\boldsymbol{z}_1||_1 & \text{for } i = 0. \end{cases}$$

Clearly $\boldsymbol{z}_2 \geq \boldsymbol{0}$ and $\boldsymbol{z}_2 \neq \boldsymbol{0}$, so we just have to check the equality constraint:

$$\begin{aligned} -z_{2,0}\boldsymbol{b}_1 + \sum_{i\geq 1} z_{2,i}\left(\boldsymbol{a}_1^{(i)} - \frac{1}{\kappa}\boldsymbol{b}_1\right) &= -\left(1 - \frac{1}{\kappa}||\boldsymbol{z}_1||_1\right)\boldsymbol{b}_1 + \sum_{i\geq 1} z_{1,i}\left(\boldsymbol{a}_1^{(i)} - \frac{1}{\kappa}\boldsymbol{b}_1\right) \\ &= -\left(1 - \frac{1}{\kappa}||\boldsymbol{z}_1||_1\right)\boldsymbol{b}_1 + \boldsymbol{b}_1 - \frac{1}{\kappa}||\boldsymbol{z}_1||_1\boldsymbol{b}_1 \\ &= 0, \end{aligned}$$

as desired.

Conversely, suppose that $\boldsymbol{z}_2$ is a solution to the system in (5.5). By definition $\boldsymbol{z}_2 \neq 0$, so it is well-defined to set

$$z_{1,i} = z_{2,i}\left(z_{2,0} + \frac{1}{\kappa}\sum_{j\geq 1} z_{2,j}\right)^{-1}$$

for all $i \geq 1$. Clearly $\boldsymbol{z}_1 \geq \boldsymbol{0}$ and $\boldsymbol{z}_1 \neq \boldsymbol{0}$, so we again need only check the equality constraint:

$$\begin{aligned} \sum_{i\geq 1} z_{1,i}\boldsymbol{a}_1^{(i)} &= \left(z_{2,0} + \frac{1}{\kappa}\sum_{j\geq 1} z_{2,j}\right)^{-1} \sum_{i\geq 1} z_{2,i}\boldsymbol{a}_1^{(i)} \\ &= \left(z_{2,0} + \frac{1}{\kappa}\sum_{j\geq 0} z_{2,j}\right)^{-1}\left(z_{2,0} + \frac{1}{\kappa}\sum_{k\geq 1} z_{2,k}\right)\boldsymbol{b}_1 \\ &= \boldsymbol{b}_1, \end{aligned}$$

where the second equality follows from the equality constraint in (5.6). This completes the proof of Lemma 5.1.1. □

## 5.2 Degeneracy and the Reduction to Certifying Boundedness

Let

$$R = \{\boldsymbol{w} \mid A_2\boldsymbol{w} \leq \mathbf{1}\}, \tag{5.7}$$

and let $\boldsymbol{a}_2^{(0)}, \ldots, \boldsymbol{a}_2^{(n)}$ be the rows of $A_2$. A feasible solution $\boldsymbol{z}$ to the system in (5.5) is a nontrivial positive combination of the rows of $A_2$ that equals the zero vector. Scaling the coefficients will give us a convex combination of the $\boldsymbol{a}_2^{(i)}$ that equals the zero vector. Since the polar polytope $R^*$ is the convex hull of the $\boldsymbol{a}_2^{(i)}$, the system in (5.5) is thus feasible if and only if the origin is contained in $R^*$.

We recall from Section 2.4 that $R$ is bounded if and only if $R^*$ contains the origin *in its interior*. By Remark 2.4.3, a feasible solution to the system in (5.5) is thus quite close to a certificate of boundedness for $R$; they differ only in the degenerate case when the origin appears on the boundary of $R^*$. In this section, we shall use a procedure similar to the $\epsilon$-perturbation technique of Charnes [11] and Megiddo and Chandrasekaran [40] to reduce solving (5.5) to solving it in the nonndegenerate case, where a solution to (5.5) is equivalent to a certificate of boundedness for $R$.

Let $A_2$ be an $m \times n$ matrix. By restricting to a subspace if necessary, we can assume that the rows of $A_2$ span $\mathbb{R}^n$ so that $R^*$ is a full-dimensional polytope, and our problem is to determine whether the origin lies in the polytope. We shall now perturb our problem slightly by pushing to origin very slightly toward the average of the $\boldsymbol{a}_2^{(i)}$. More precisely, we shall seek a feasible solution to the system

$$\begin{aligned} A_2^T\left(\boldsymbol{q} - \epsilon\left(\textstyle\sum_i q_i/m\right)\mathbf{1}\right) &= \boldsymbol{0} \\ \boldsymbol{q} \geq \boldsymbol{0}, \boldsymbol{q} &\neq \boldsymbol{0}, \end{aligned} \tag{5.8}$$

where $\epsilon = 1/2^{\text{poly}(m)\cdot L}$ with a sufficiently large polynomial in the exponent. We can write this in the same form as the system in (5.5) by letting $A_3$ be the matrix whose $i^{\text{th}}$ row is given by

$$\boldsymbol{a}_3^{(i)} = \boldsymbol{a}_2^{(i)} - \frac{\epsilon}{m}\sum_i \boldsymbol{a}_2^{(i)} \tag{5.9}$$

and considering the system

$$\begin{aligned} A_3^T\boldsymbol{q} &= \boldsymbol{0} \\ \boldsymbol{q} \geq \boldsymbol{0},\ \boldsymbol{q} &\neq \boldsymbol{0}. \end{aligned}$$

We claim that this yields a polynomial-time reduction to the nondegenerate case. This follows from the following four properties of the system in (5.8):

**Property 1:** Given the system in (5.5), we can construct the system in (5.8) in polynomial time.

**Property 2:** If (5.5) is feasible then (5.8) has a solution whose coordinates are all strictly positive.

**Property 3:** If (5.5) is infeasible then (5.8) is infeasible.

**Property 4:** Given a solution to (5.8), one can recover a solution to (5.5) in polynomial time.

*Proof of Property 1.* This follows immediately from the description of the system in (5.8) and the fact that the bit-length of $\epsilon$ is polynomial in $L$. □

*Proof of Property 2.* Let $\overline{\boldsymbol{q}}$ be a feasible point for (5.5), so that

$$A_2^T \overline{q} = \boldsymbol{0}.$$

Let

$$\widetilde{\boldsymbol{q}} = \overline{\boldsymbol{q}} + \frac{\epsilon \sum_i \overline{q}_i}{m(1-\epsilon)} \mathbf{1}.$$

We note that $\widetilde{\boldsymbol{q}}$ is a feasible solution to (5.8):

$$\begin{aligned}
A_2^T \left( \widetilde{\boldsymbol{q}} - \epsilon \left( \sum_i \widetilde{q}_i / m \right) \mathbf{1} \right) &= A_2^T \left( \left( \overline{\boldsymbol{q}} + \frac{\epsilon \sum_i \overline{q}_i}{m(1-\epsilon)} 1 \right) - \frac{\epsilon}{m} \left( \sum_i \overline{q}_i + \frac{\epsilon m \sum_i \overline{q}_i}{m(1-\epsilon)} \right) \mathbf{1} \right) \\
&= A_2^T \overline{\boldsymbol{q}} + A_2^T \left( \left( \frac{\epsilon \sum_i \overline{q}}{m(1-\epsilon)} - \frac{\epsilon}{m} \left( 1 + \frac{\epsilon}{1-\epsilon} \right) \sum_i \overline{q}_i \right) \mathbf{1} \right) \\
&= \boldsymbol{0} + \left( \frac{\epsilon}{m(1-\epsilon)} - \frac{\epsilon}{m} \frac{(1-\epsilon)+\epsilon}{1-\epsilon} \right) \left( \sum_i \overline{q}_i \right) A_2^T \mathbf{1} \\
&= \boldsymbol{0} + 0 \left( \sum_i \overline{q}_i \right) A_2^T \mathbf{1} \qquad (5.10) \\
&= \boldsymbol{0},
\end{aligned}$$

as desired. Since all of the coefficients of $\widetilde{q}$ are strictly positive, this establishes Property 2. □

*Proof of Property 3.* Let $\boldsymbol{a}_3^{(i)}$ be as in equation (5.9). The system in (5.5) is feasible if and only if the origin is contained in the convex hull $R^*$ of the $\boldsymbol{a}_2^{(i)}$, whereas the

system in (5.8) is feasible if and only if the origin is contained in the convex hull of the $\boldsymbol{a}_3^{(i)}$.

Suppose that (5.5) is infeasible; we show that this implies that (5.8) is infeasible as well. To this end, let $\boldsymbol{p} \in R^*$ be the point on $R^*$ that is closest to the origin. The point $\boldsymbol{p}$ lies on the boundary of $R^*$, so there exists some collection of $n$ of the $\boldsymbol{a}_2^{(i)}$ that spans a nondegenerate simplex $\Delta$ that contains $p$. Without loss of generality, let this collection consist of $\boldsymbol{a}_2^{(1)}, \ldots, \boldsymbol{a}_2^{(n)}$. Let

$$\overline{\Delta} = \mathbf{conv}(\Delta, \boldsymbol{0}).$$

The $n$-dimensional volume of $\overline{\Delta}$ equals $1/n$ times the $(n-1)$-dimensional volume of $\Delta$ times the orthogonal distance from the hyperplane spanned by $\Delta$ to the origin. We thus have

$$||\boldsymbol{p}||_2 \geq \frac{n \cdot \mathrm{vol}_n(\overline{\Delta})}{\mathrm{vol}_{n-1}(\Delta)}.$$

If $M_1$ is the $n \times n$ matrix whose $i^{\text{th}}$ row equals $\boldsymbol{a}_2^{(i)T}$, and $M_2$ is the $(n-1) \times n$ matrix whose $i^{\text{th}}$ row equals $\boldsymbol{a}_2^{(i)} - \boldsymbol{a}_2^{(n)}$, we can expand this as

$$\begin{aligned} ||\boldsymbol{p}||_2 &\geq \frac{n \cdot \mathrm{vol}_n(\overline{\Delta})}{\mathrm{vol}_{n-1}(\Delta)} \\ &= \frac{n \cdot (1/n!)\sqrt{\det\left(M_1^T M_1\right)}}{(1/(n-1)!)\sqrt{\det\left(M_2^T M_2\right)}} \\ &= \sqrt{\frac{\det\left(M_1^T M_1\right)}{\det\left(M_2^T M_2\right)}}. \end{aligned} \tag{5.11}$$

All of entries in $M_1$ and $M_2$ have bit-lengths that are bounded above by $L$, so the numerator and denominator of the fraction under the square root can both be written with $\mathrm{poly}(m) \cdot L$ bits, and thus so can the entire fraction. Since we have assumed that $||\boldsymbol{p}||_2 \neq 0$, this implies a $1/2^{\mathrm{poly}(m) \cdot L}$ lower bound on $||\boldsymbol{p}||_2$.

We thus have a lower bound $\ell$ on the distance between the convex hull of the $\boldsymbol{a}_2^{(i)}$ and the origin. If we displace each $\boldsymbol{a}_2^{(i)}$ by less than $\ell$, no convex combination of the $\boldsymbol{a}_2^{(i)}$ can move by more than $\ell$, so the perturbed polytope will not contain the origin. The distance between $\boldsymbol{a}_2^{(i)}$ and $\boldsymbol{a}_3^{(i)}$ is at most

$$\frac{\epsilon}{m}\left|\left|\left(\sum_i \boldsymbol{a}_2^{(i)}\right)\right|\right|_2,$$

so, as long as

$$\epsilon < \frac{\ell m}{\left\| \left( \sum_i \boldsymbol{a}_2^{(i)} \right) \right\|_2} = \Omega\left( \frac{1}{2^{\text{poly}(m) \cdot L}} \right),$$

the convex hull of the $\boldsymbol{a}_3^{(i)}$ will not contain the origin. This implies the infeasibility of (5.8), as desired. □

*Proof of Property 4.* Given any solution to (5.8), standard techniques allow one to recover, in polynomial time, a solution at which exactly $n$ of the $q_i$ are nonzero and for which the corresponding $\boldsymbol{a}_3^{(i)}$ are linearly independent [45]. Scaling so that the coefficients add to 1, this shows that the origin is contained inside the simplex spanned by $n$ of the $\boldsymbol{a}_3^{(i)}$. The proof of Property 3 shows that the simplex spanned by the corresponding $\boldsymbol{a}_2^{(i)}$ will also contain the origin, i.e., that the origin can be written as a convex combination of the corresponding $\boldsymbol{a}_2^{(i)}$. We can find the coefficients of this convex combination in polynomial time by solving a linear system, and this is our desired solution. □

It thus suffices to be able to find a certificate of boundedness for the polytope described in (5.7). This is equivalent to proving that

$$A_2 \boldsymbol{w} \leq \boldsymbol{b}_2 \tag{5.12}$$

is bounded for *any* $\boldsymbol{b}_2 > \boldsymbol{0}$, since the choice of the vector $\boldsymbol{b}_2$ does not affect whether the polytope is bounded. (We require $\boldsymbol{b}_2 > \boldsymbol{0}$ in order to guarantee that the resulting system is feasible.) By solving this system with a randomly chosen right-hand side vector we can solve system (1.1) while avoiding the combinatorial complications of the feasible set of (1.1).

In our algorithm, we will certify boundedness of (1.1) by finding the vertices minimizing and maximizing some objective function. Provided that the system is non-degenerate, which it is with high probability under our choice of right-hand sides, this can be converted into a solution to (5.5).

# Chapter 6

# Our Algorithm

Our bound from Theorem 4.1.2 suggests a natural algorithm for certifying the boundedness of a linear program of the form given in (5.12): set each $b_i$ to be $1+r_i$, where $r_i$ is an exponential random variable, pick a random objective function $c$ and a random two-dimensional subspace containing it, and then use the shadow-vertex method with the given subspace to maximize and minimize $c$.

In order to make this approach into a polynomial-time algorithm, there are two difficulties that we must surmount:

1. To use the shadow-vertex method, we need to start with some vertex that appears on the boundary of the shadow. If we just pick an arbitrary shadow plane, there is no obvious way to find such a vertex.

2. Theorem 4.1.2 bounds the expected shadow size of the vertices of bounded norm in polytopes with perturbed right-hand sides, whereas the polytope that we are given may have vertices of exponentially large norm. If we naively choose our perturbations, objective function, and shadow plane as if we were in a coordinate system in which all of our vertices had bounded norm, the distribution of vertices that appear on the shadow may be very different, and we have no guarantees about the expected shadow size.

We address the first difficulty by constructing an artificial vertex at which to start our simplex algorithm. To address the second difficulty, we start out by choosing our random variables from the naive distributions. If this doesn't work, we iteratively use information about how it failed to improve the probability distributions from which we sample and try again.

## 6.1 Constructing a Starting Vertex

In order to use the shadow-vertex method on a polytope $P$, we need a shadow plane $\boldsymbol{S}$ and a vertex $\boldsymbol{v}$ that appears on the boundary of the shadow. One way to obtain such a pair is to pick any vertex $\boldsymbol{v}$, randomly choose (from some probability distribution) an objective function $\boldsymbol{c}$ optimized by $\boldsymbol{v}$, let $\boldsymbol{u}$ be a uniformly random unit vector, and set $\boldsymbol{S} = \text{span}(\boldsymbol{c}, \boldsymbol{u})$.

However, to apply the bound on the shadow size given by Theorem 4.2.2, we need to choose $\boldsymbol{c}$ to be a $\rho$-perturbation of some vector. For such a $\boldsymbol{c}$ to be likely to be optimized by $\boldsymbol{v}$, we need $\boldsymbol{v}$ to optimize a reasonably large ball of objective functions. To guarantee that we can find such a $\boldsymbol{v}$, we create one. That is, we add constraints to our polytope to explicitly construct an artificial vertex with the desired properties. (This is similar to the "Phase I" approaches that have appeared in some other simplex algorithms.)

Suppose for now that the polytope $\{\boldsymbol{x} | A\boldsymbol{x} \leq \mathbf{1}\}$ is $k$-round. Construct a modified polytope $P'$ by adding $d$ new constraints, $\{\overline{\boldsymbol{w}}_i^T \boldsymbol{x} \leq 1, i = 1, \ldots, d\}$, where

$$\boldsymbol{w}_i = -\Big(\sum_j \boldsymbol{e}_j\Big) + \sqrt{d}\boldsymbol{e}_i/3k^2,$$

and $\overline{\boldsymbol{w}}_i = \boldsymbol{w}_i/(2\,||\boldsymbol{w}_i||)$. Let $\boldsymbol{x}_0$ be the vertex at which $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d$ are all tight. Furthermore, let $\boldsymbol{c}$ be a $\rho$-perturbation of the vector $\mathbf{1}/\sqrt{d}$, with $\rho = 1/6dk^2$, and let $\boldsymbol{x}_1$ be the vertex at which $\boldsymbol{c}$ is maximized. We can prove:

**Lemma 6.1.1.** *The following three properties hold with high probability. Furthermore, they remain true with probability* $1 - (d+2)e^{-n}$ *if we perturb all of the right-hand sides of the constraints in* $P'$ *by an exponential random variable of expectation* $\lambda = 1/n$.

1. *The vertex* $\boldsymbol{x}_0$ *appears on* $P'$,
2. $-\boldsymbol{c}$ *is maximized at* $\boldsymbol{x}_0$, *and*
3. *None of the constraints* $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d$ *is tight at* $\boldsymbol{x}_1$.

*Proof.* This follows from Lemma 7.0.1 and bounds on tails of exponential random variables. □

Set

$$k := 16d + 1 \quad \text{and} \quad s := 4 \cdot 10^7\, d^{9/2} n.$$

Let $\boldsymbol{S} = \text{span}(\boldsymbol{c}, \boldsymbol{u})$, where $u$ is a uniform random unit vector. If $P$ is $k$-round, then by Lemma 6.1.1 and Theorem 4.1.2 we can run the shadow vertex method on $P'$

with shadow plane $\boldsymbol{S}$ and starting at vertex $\boldsymbol{x}_0$, and we will find the vertex $\boldsymbol{x}_1$ that maximizes $\boldsymbol{c}$ within $s$ steps, with probability at least $1/2$. Since none of the $\boldsymbol{w}_i$ are tight at $\boldsymbol{x}_1$, $\boldsymbol{x}_1$ will also be the vertex of the original polytope $P$ that maximizes $\boldsymbol{c}$.

This gives us the vertex $\boldsymbol{x}_1$ of $P$ that maximizes $\boldsymbol{c}$. We can now run the shadow vertex method again on $P$ using the same shadow plane. This time, we *start* at $\boldsymbol{x}_1$ and find the vertex that maximizes $-\boldsymbol{c}$. We are again guaranteed to have an expected polynomial-sized shadow, so this will again succeed with high probability. This will give us a pair of vertices that optimize $\boldsymbol{c}$ and $-\boldsymbol{c}$, from which we can compute our desired certificate of boundedness. It just remains to deal with polytopes that are not $k$-round position.

## 6.2 Polytopes that are not $k$-Round

In this section, we shall present and analyze our general algorithm that deals with polytopes that may not be $k$-round. The pseudocode for this algorithm appears at the end of the section, on page 46.

We first observe that for every polytope there exists an affine change of coordinates (*i.e.*, a translation composed with a change of basis) that makes it $d$-round [5]. An affine change of coordinates does not change the combinatorial structure of a polytope, so this means that there exists *some* probability distribution on $\boldsymbol{b}$ and $\boldsymbol{S}$ for which the shadow has polynomial expected size. We would like to sample $\boldsymbol{b}$ and $\boldsymbol{S}$ from these probability distributions and then pull the result back along the change of coordinates. Unfortunately, we don't know an affine transformation that makes our polytope $k$-round, so we are unable to sample from these distributions.

Instead, we shall start out as we would in the $k$-round case, adding in artificial constraints $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d$, and choosing an objective function and shadow plane as in Section 6.1. By Theorem 4.2.2, running the shadow-vertex method for $s$ steps will yield one of two results with probability at least $1/2$:

1. It will find the optimal vertex $\boldsymbol{x}_1$, or
2. It will find a vertex $\boldsymbol{y}$ of norm at least $2k$.

In the first case, we can proceed just as in the $k$-round case and run the shadow-vertex method a second time to optimize $-\boldsymbol{c}$, for which we will have the same two cases.

In the second case, we have not found the optimal vertex, but we have with high probability learned a point of large norm inside our polytope. We can use this point to change the probability distributions from which we draw our random variables and then start over. This changes our randomized pivot rule on the graph of potential vertices of our polytope, hopefully putting more probability mass on short paths from

the starting vertex to the optimum. We shall show that, with high probability, we need only repeat this process a polynomial number of times before we find a right-hand side and shadow plane for which the shadow-vertex method finds the optimum.

Our analysis rest upon the following geometric lemma, proved in Chapter 7:

**Lemma 6.2.1.** *Let $B \subseteq \mathbb{R}^d$ be the unit ball, let $P$ be a point at distance $S$ from the origin, and let $C = \text{conv}(B, P)$ be their convex hull. If $S \geq 16d + 1$, then $C$ contains an ellipse of volume at least twice that of $B$, having $d-1$ semi-axes[1] of length $1 - 1/d$ and one semi-axis of length at least 8 centered at the point of distance 7 from the origin in the direction of $P$.*

We remark that the number of times that we have to change probability distributions depends on the bit-length of the inputs, and that this is the only part of our algorithm in which this is a factor. Otherwise, the execution of our algorithm is totally independent of the bit-length of the inputs.

**Theorem 6.2.2.** *If each entry of the vectors $\boldsymbol{a}_i$ is specified using $L$ bits, then* CHECKBOUNDEDNESS() *either produces a certificate that its input is bounded or that it is unbounded within $O(n^3 L)$ iterations, with high probability.*

*Proof.* It will be helpful to think of the input to CHECKBOUNDEDNESS() as being the polytope $\{\boldsymbol{x} \mid \boldsymbol{a}_i^T x \leq 1 \, \forall i\}$ instead of just the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. We can then talk about running this algorithm on an arbitrary polytope $\{\boldsymbol{x} \mid \alpha_i^T x \leq \tau_i \, \forall i\}$ by rewriting this polytope as $\{\boldsymbol{x} \mid (\alpha_i/\tau_i)^T x \leq 1 \, \forall i\}$.

With this notation, it is easy to check that running an iteration of the **Repeat** loop on a polytope $P$ with $\boldsymbol{Q} = \boldsymbol{Q}_0$ and $\boldsymbol{r} = \boldsymbol{r}_0$ is equivalent to running the same code on the polytope $\boldsymbol{Q}_0(P + \boldsymbol{r}_0)$ with $\boldsymbol{Q} = \text{Id}$ and $\boldsymbol{r} = 0$. The update step at the end of the algorithm can therefore be thought of as applying an affine change of coordinates to the input and then restarting the algorithm.

If $\boldsymbol{Q} = \text{Id}_n$ and $\boldsymbol{r} = 0$, the argument from Section 6.1 proves that the first iteration of the **Repeat** loop will either prove boundedness, prove unboundedness, or find a point with norm at least $k$ with probability at least $1/2$. In either of the first two cases, the algorithm will have succeeded, so it suffices to consider the third.

If a point $\boldsymbol{y}$ is in the polytope $P' = \{\boldsymbol{x} \mid A\boldsymbol{x} \leq \boldsymbol{b}\}$, the point $\boldsymbol{y}/2$ will be in the polytope $P = \{x \mid A\boldsymbol{x} \leq \boldsymbol{1}\}$ with probability at least $1 - ne^{-n}$. This guarantees that $P$ contains a point of norm at least $k$. Since $P$ contains the unit ball, Lemma 6.2.1 implies that $P$ contains an ellipse of volume at least twice that of the unit ball. The

[1] *If an ellipsoid $E$ is given as the set $E = \{x \mid x^T Q^{-1} x \leq 1\}$, where $Q$ is a symmetric, positive definite matrix, then the* semi-axes *of $E$ have lengths equal to the the eigenvalues of $Q$. For example, the semi-axes of the sphere are all of length 1.*

update step of our algorithm identifies such an ellipse and scales and translates so that it becomes the unit ball, and it then restarts with this new polytope as its input. This new polytope has at most half the volume of the original polytope.

All the vertices of the original polyhedron are contained in a ball of radius $2^{O(n^2L)}$, where $L$ is the maximum bit-length of any number in the input, and so their convex hull has volume at most $2^{O(n^3L)}$ times that of the unit ball [26]. Each iteration of the algorithm that finds a point of norm at least $k$ decreases the volume of $P$ by a factor of at least 2. All of the polytopes that we construct contain the unit ball, so this can occur at most $O(n^3L)$ times. This guarantees that the **Repeat** loop finds an answer after a $O(n^3L)$ iterations with high probability, as desired. □

While the algorithm requires samples from the exponential distribution and uniform random points on the unit sphere, it is not difficult to show that it suffices to use standard discretizations of these distributions of bit-length polynomial in $n$ and $d$.

**Algorithm 6.2.1:** CHECKBOUNDEDNESS($\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$)

**Require** each $\boldsymbol{a}_i$ has norm at most 1.
Set $k = 16d + 1$, $\lambda = 1/n$, $\rho = 1/6dk^2$
$s = 4 \cdot 10^7\, d^{9/2} n$, and $\overline{\boldsymbol{w}_i}$ as described in text;
Initialize $\boldsymbol{Q} := \mathrm{Id}_n$, $\boldsymbol{r} := \boldsymbol{0}$;
**Repeat** until you return an answer
{
Construct constraints for starting corner:
$\boldsymbol{a}_{n+i} := \boldsymbol{Q}^T \overline{\boldsymbol{w}_i} / (1 - \overline{\boldsymbol{w}_i} \cdot (\boldsymbol{Q}\boldsymbol{r}))$ for $i = 1, \ldots, d$;
$b_i := (1 + \beta_i)(1 + \boldsymbol{a}_i^T \boldsymbol{r})$ for $i = 1, \ldots n + d$, (1)
$\beta_i$ exponential random vars with expectation $\lambda$;
Set starting corner $\boldsymbol{x}_0 :=$ point where $\boldsymbol{a}_i^T \boldsymbol{x}_0 = b_i$
for $i = n + 1, \ldots, n + d$;
**If** $\boldsymbol{x}_0$ violates $\boldsymbol{a}_i^T \boldsymbol{x}_0 \leq b_i$ for any $i$, go back to (1)
and generate new random variables;
$c := \boldsymbol{Q}^T \gamma$, with $\gamma$ a $\rho$-perturbation of $\boldsymbol{1}/\sqrt{d}$;
Shadow plane $\boldsymbol{S} := \mathrm{span}(\boldsymbol{c}, \boldsymbol{Q}^T \boldsymbol{u})$,
with $\boldsymbol{u}$ a uniformly random unit vector;
Run SHADOWVERTEX($(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n+d}), \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{S}, \boldsymbol{x}_0, s$) :
{
**If** returns **unbounded then**
**return (unbounded)**;
**If** returns (**fail**, $\boldsymbol{y}_0$) **then**
set $\boldsymbol{y} := \boldsymbol{y}_0$ and go to (3);
**If** returns (**opt**, $\boldsymbol{v}_0$) **then**
set $\boldsymbol{v} := \boldsymbol{v}_0$ and continue to (2);
}
Run SHADOWVERTEX($(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n), \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{S}, \boldsymbol{v}, s$) : (2)
{
**If** returns **unbounded then**
**return (unbounded)**;
**If** returns (**fail**, $\boldsymbol{y}_0$) **then**
set $\boldsymbol{y} := \boldsymbol{y}_0$ and go to (3);
**If** returns (**opt**, $\boldsymbol{v}_0$) **then**
set $\boldsymbol{v}' := \boldsymbol{v}_0$ and **return** $(\boldsymbol{v}, \boldsymbol{v}')$;
}
Update $\boldsymbol{Q}$ and $\boldsymbol{r}$: (3)
{
**If** $||\boldsymbol{Q}(\boldsymbol{y} + \boldsymbol{r})|| \leq 2k$ **then**
don't change $\boldsymbol{Q}$ or $\boldsymbol{r}$ .
**else**
{
Set $\boldsymbol{M}$ := the matrix that scales down
$Q(\boldsymbol{y} + \boldsymbol{r})$ by factor of 8 and scales
vectors in orthogonal complement up
by factor of $1 - 1/d$;
$\boldsymbol{Q} := \boldsymbol{M}\boldsymbol{Q}$;
$\boldsymbol{r} := \boldsymbol{r} + 7Q(\boldsymbol{y} + \boldsymbol{r})/||Q(\boldsymbol{y} + \boldsymbol{r})||$;
}
}

## 6.3 Towards a Strongly Polynomial-Time Algorithm for Linear Programming?

While it is usually best to avoid the risky business of predicting yet-unproven results, it is worth briefly noting that we believe these results to be encouraging progress towards finding a strongly polynomial-time algorithm for linear programming.

First of all, these results provide significant geometric insights into the structure of linear programming and polytope theory, and they provide a new approach to constructing algorithms for linear programming. Our algorithm proceeds almost entirely in strongly polynomial time, and it runs in strongly polynomial time for a large class of linear programs. The only part of the algorithm that is not strongly polynomial is the outer loop in which we alter the various probability distributions. It seems quite plausible that this dependence can be eliminated by a slightly more clever variant of our algorithm.

Furthermore, our methods suggest a wide variety of similar approaches. While the shadow-vertex method was the easiest simplex method to analyze, it may well be the worst one to use when searching for a strongly polynomial-time algorithm. The dependence of the running-time of the algorithm on the bit-length arises from the linear program being given initially in a "bad" coordinate system. The shadow-vertex method is perhaps, among reasonable pivot rules, the one that depends the most adversarially upon the ambient coordinate system. If one could obtain a similar analysis of the behavior on linear programs with perturbed right-hand sides of a less coordinate-dependent pivot rule, such as RANDOM-EDGE (see [23], for example), it is quite possible that the dependence on the bit-length would disappear.

# Chapter 7

# Geometric Lemmas for Algorithm's Correctness

**Lemma 7.0.1.** *Let $P$ be a $k$-round polytope, let $c$ and $q$ be unit vectors, and let*

$$v = \operatorname*{argmax}_{x \in P} c \cdot x$$

*be the vertex of $P$ at which $c \cdot x$ is maximized. If $c \cdot q \leq -(2k^2 - 1)/2k^2$, then $v \cdot q \leq 0$.*

*Proof.* We first note that

$$||q + c||^2 = ||q||^2 + ||c||^2 + 2(c \cdot q) \leq 2 - \frac{2k^2 - 1}{k^2} = \frac{1}{k^2},$$

so $||q + c|| \leq 1/k$. The fact that $P$ is contained in $B(0, k)$ implies that $||v|| \leq k$, and the fact that $P$ contains the unit ball implies that

$$v \cdot c = \max_{x \in P} c \cdot x \geq 1.$$

We therefore have

$$q \cdot v = -c \cdot v + (q + c) \cdot v \leq -1 + ||q + c||||v|| \leq 0,$$

as desired. □

We now prove some geometric facts that will be necessary for the analysis of our algorithm. We first prove a two-dimensional geometric lemma. We then use this to prove a higher-dimensional analogue, which is the version that we shall actually use to analyze our algorithm.

## 7.1 2-Dimensional Geometry Lemma

In this section, we prove a lemma about the two-dimensional objects shown in Figure 7-1. In this picture, $C$ is the center of a circle $\mathbf{C}$ of radius 1. $P$ is a point somewhere along the positive $x$-axis, and we have drawn the two lines tangent to the circle through $P$, the top one of which we have labeled $L$. $E$ is the center of an axis-parallel ellipse $\mathbf{E}$ with horizontal semi-axis $M \geq 1$ and vertical semi-axis $m \leq 1$. The ellipse is chosen to be a maximal ellipse contained in the convex hull of the circle and $P$. Furthermore, let $S$ be the distance from $C$ to $P$, and let $Q = (1 - m^2)/2$.

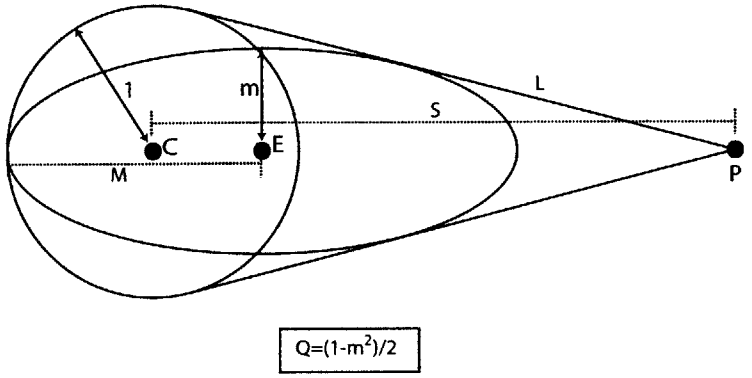


Figure 7-1: The geometric objects considered in Lemma 7.1.1

**Lemma 7.1.1.** *With the definitions above,*

$$M = Q(S - 1) + 1.$$

*Proof.* Without loss of generality, let $E$ be the origin. The circle and ellipse are mutually tangent at their leftmost points on the $x$-axis, so $\mathbf{C}$ is at $(-M + 1, 0)$, and $P$ is therefore at $(S - M + 1, 0)$. Let

$$\ell = \left(\frac{1}{S}, \sqrt{1 - \frac{1}{S^2}}\right),$$

and let $L$ be the line given by

$$L = \left\{(x, y) \mid \ell \cdot (x, y) = \frac{S - M + 1}{S}\right\}.$$

We claim that $L$ has the following three properties, as shown in Figure 7-1:

1. $L$ passes through $P$.
2. $L$ is tangent to $\mathbf{C}$.
3. If we take the major semi-axis $M$ of the ellipse $\mathbf{E}$ to be $Q(S-1)+1$, then $L$ is tangent to $\mathbf{E}$.

Establishing these properties would immediately imply Lemma 7.1.1, so it suffices to check them one by one.

1. This follows by direct computation—we simply note that the point $P = (S - M + 1, 0)$ satisfies the equation for $L$.
2. It suffices to show that the distance from the point $C$ to the line $L$ is exactly 1. Since $\mathcal{L}$ is the unit normal to $L$, it suffices to check that
$$\ell \cdot C = \left(\frac{S-M+1}{S}\right) - 1 = \frac{-M+1}{S},$$
which again follows by direct computation.
3. Let
$$\mathcal{L} = (\mathcal{L}_x, \mathcal{L}_y) = \frac{S}{S-M+1}\ell = \left(\frac{1}{S-M+1}, \frac{\sqrt{S^2-1}}{S-M+1}\right),$$
so that $L = \{(x,y) \mid \mathcal{L} \cdot (x,y) = 1\}$. When expressed in this form, $L$ will be tangent to $E$ if and only if $\mathcal{L}_x^2 M^2 + \mathcal{L}_y^2 m^2 = 1$. This can be verified by plugging in $M = Q(S-1)+1$ and $Q = (1-m^2)/2$, and then expanding the left-hand side of the equation.

□

## 7.2 High-Dimensional Geometry Lemma

**Lemma 7.2.1.** *Let $B \subseteq \mathbb{R}^d$ be the unit ball, let $P$ be a point at distance $S$ from the origin, and let $C = \mathrm{conv}(B, P)$ be their convex hull. For any $m \leq 1$, $C$ contains an ellipsoid with $(d-1)$ semi-axes of length $m$ and one semi-axis of length $(1-m^2)(S-1)/2 + 1$.*

*Proof.* Without loss of generality, take $P = (S, 0, \ldots, 0)$. Consider an axis-parallel ellipsoid $E$ with the axes described in the above theorem, with its distinct axis parallel to $e_1$, and translated so that it is tangent to $B$ at $(-1, 0, \ldots, 0)$.

We assert that $E$ is contained in $C$. It suffices to check the containment when we intersect with an arbitrary 2-dimensional subspace containing 0 and $P$. In this case, we have exactly the setup of Lemma 7.1.1, and our result follows immediately. □

*of Lemma 6.2.1.* If we set $m = 1 - 1/d$, then Lemma 7.2.1 guarantees that the length of the longer semi-axis of the ellipse will be at least

$$\left(1 - \left(1 - \frac{1}{d}\right)^2\right) \frac{16d}{2} \geq 8.$$

So, the ratio of the volume of the unit ball to the ellipse is at least

$$\begin{aligned} V/\mathrm{vol}(B) &\geq \left(1 - \frac{1}{d}\right)^{d-1} 8 \\ &\geq \frac{8}{4} = 2. \end{aligned}$$

□

# Part II

# Spectral Partitioning, Eigenvalue Bounds, and Circle Packings for Graphs of Bounded Genus

# Chapter 8

# Background in Graph Theory and Spectral Partitioning

In this chapter we provide the basic definitions and results from graph theory and spectral partitioning that we shall require in the sequel.

## 8.1 Graph Theory Definitions

Throughout the remainder of this part of the thesis, let $G = (V, E)$ be a finite, connected, undirected graph with $n$ vertices, $m$ edges, and no loops. In this section, we shall define two objects associated to $G$: its *Laplacian*, and its *genus*.

Let the *adjacency matrix* $A(G)$ be the $n \times n$ matrix whose $(i,j)^{\text{th}}$ entry equals 1 if $(i,j) \in E$, and equals 0 otherwise. Let $D(G)$ be the $n \times n$ diagonal matrix whose $i^{\text{th}}$ diagonal entry equals the degree of the $i^{\text{th}}$ vertex of $G$.

**Definition 8.1.1.** The *Laplacian* $L(G)$ is the $n \times n$ matrix given by

$$L(G) = D(G) - A(G).$$

Since $L(G)$ is symmetric, it is guaranteed to have an orthonormal basis of real eigenvectors and exclusively real eigenvalues. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $L(G)$, and let $v_1, \ldots, v_n$ be a corresponding orthonormal basis of eigenvectors. For any $G$, the all-ones vector will be an eigenvector of eigenvalue 0. It is not difficult to see that all of the other eigenvalues will always be nonnegative, so that $v_1 = (1, \ldots, 1)^T$, and $\lambda_1 = 0$.

There has been a great deal of work relating the eigenvalues of $L(G)$ to the structure of $G$. In the present paper, we shall concern ourselves exclusively with $\lambda_2$, also known as the *algebraic connectivity* or *Fiedler value* of $G$. We call the vector $v_2$ the

*Fiedler vector* of $G$. As we shall see in Section 8.2, the Fiedler value of a graph is closely related to how well connected the graph is.

A different measure of the connectivity of a graph is provided by its *genus*, which measures the complexity of the simplest orientable surface on which the graph can be embedded so that none of its edges cross. Standard elementary topology provides a full classification of the orientable surfaces without boundary. Informally, they are all obtained by attaching finitely many "handles" to the sphere, and they are fully topologically classified (*i.e.*, up to homeomorphism) by the number of such handles. This number is called the *genus* of the surface. The genus 0, 1, 2, and 3 surfaces are shown in Figure 8-1.

Figure 8-1: The surfaces of genus 0, 1, 2, and 3.

**Definition 8.1.2.** The *genus* $g$ of a graph $G$ is the smallest integer such that $G$ can be embedded on a surface of genus $g$ without any of its edges crossing one another.

In particular, a planar graph has genus 0. By making a separate handle for each edge, it is easy to see that $g = O(m)$, where $m$ is the number of edges in $G$.

Using these definitions, we can now state our main technical result:

**Theorem 8.1.3.** *Let $G$ be a graph of genus $g$ and bounded degree. Its Fiedler value obeys the inequality*

$$\lambda_2 \leq O(g/n),$$

*and this is asymptotically tight.*

The constant in this bound depends on the degree of the graph. The proof that we provide yields a polynomial dependence on the degree, but no effort is made to optimize this polynomial. Finding the optimal such dependence is an interesting open question.

## 8.2 Spectral Partitioning

We recall that a *partition* of a graph $G$ is a decomposition $V = A \cup \overline{A}$ of the vertices of $G$ into two disjoint subsets. For such a partition, we let $\delta(A)$ be the set of edges

$(i, j)$ such that $i \in A$ and $j \in \overline{A}$, and we call $|\delta(A)|$ the *cut size* of our partition. The *ratio* of our partition is defined to be

$$\phi(A) = \frac{|\delta(A)|}{\min(|A|, |\overline{A}|)}.$$

If our partition splits the graph into two sets that differ in size by at most one, we call it a *bisection.*

Spectral methods aim to use the Fiedler vector to find a partition of the graph with a good ratio. A theorem that begins to address why these work was proven by Mihail and restated in a more applicable form by Spielman and Teng:

**Theorem 8.2.1** ([41, 47]). *Let $G$ have maximum degree $\Delta$. For any vector $x$ that is orthogonal to the all-ones vector, there is a value $s$ so that the partition of $G$ into $\{i : x_i \leq s\}$ and $\{i : x_i > s\}$ has ratio at most*

$$\sqrt{2\Delta \frac{x^T L(G) x}{x^T x}}.$$

If $x$ is an eigenvector of $L(G)$, the fraction $\frac{x^T L(G) x}{x^T x}$ is equal to its eigenvalue. So, if we find the eigenvector with eigenvalue $\lambda_2$, we will thus quickly be able to find a partition of ratio $\sqrt{2\Delta\lambda_2}$. By Theorem 8.1.3, finding the second eigenvector of the Laplacian thus allows us to find a partition of ratio $O(\sqrt{g/n})$ for a graph of bounded degree. There is no guarantee that this partition has a similar number of vertices in each of the two sets. However, a theorem of Lipton and Tarjan [39] implies that a simple method based on repeated application of this algorithm can be used to give a bisector of size $O(\sqrt{gn})$.

For every $g$, Gilbert, Hutchinson, and Tarjan exhibited a class of bounded degree graphs that have no bisectors smaller than $O(\sqrt{gn})$ [25]. This implies that our algorithm gives the best results possible, in general. Furthermore, it establishes the asymptotic tightness of our eigenvalue bound, as a smaller bound would show that every genus $g$ graph has a partition of size $o(\sqrt{gn})$.

Putting all of this together yields our main algorithmic result:

**Theorem 8.2.2.** *Let $G$ be a genus $g$ graph of bounded maximum degree. There is a polynomial time algorithm that produces cuts of ratio $O(\sqrt{g/n})$ and vertex bisectors of size $O(\sqrt{gn})$ in $G$, and both of these values are optimal.*

All that remains of the proof of Theorem 8.2.2 is the eigenvalue bound set forth in Theorem 8.1.3, which is the goal of the remainder of this paper.

# Chapter 9

# Outline of the Proof of the Main Technical Result

The proof of Theorem 8.1.3 necessitates the introduction of a good deal of technical machinery. Before launching into several pages of definitions and background theorems, we feel that a brief roadmap of where we're going will be helpful.

The basic motivation for our approach comes from an observation made by Spielman and Teng [47]. They noted that one can obtain bounds on the eigenvalues of a graph $G$ from a nice representation of $G$ on the unit sphere in $\mathbb{R}^3$ known as a *circle packing* for $G$. This is a presentation of the graph on the sphere so that the vertices are the centers of a collection of circles, and the edges between vertices correspond to tangencies of their respective circles, as shown in Figure 10-1. Only planar graphs can be embedded as such if we require the circles to have disjoint interiors. However, if we allow the circles to overlap, as shown in Figure 10-2, we can represent nonplanar graphs as well. This will give rise to a weaker bound in which the eigenvalue bound is multiplied by the maximum number of circles containing a given point (*i.e.*, the number of layers of circles on the sphere).

There is a well developed theory of circle packings, both on the sphere and on higher genus surfaces. The portions of it that we shall use will tell us two main things:

1. We can realize our graph as a circle packing of circles with disjoint interiors on some genus $g$ surface.

2. The theory of discrete circle packings can be thought of as a discrete analogue of classical complex function theory, and many of the results of the latter carry over to the former.

In classical complex analysis, you can put a complex analytic structure on a genus $g$ surface to obtain a Riemann surface. Any genus $g$ Riemann surface has a map to the

sphere that is almost everywhere $k$-to-one for $k = O(g)$, with only $O(g)$ bad points at which this fails. With this as motivation, we shall try to use the representation of $G$ as a circle packing on a genus $g$ surface to obtain a representation of it as a circle packing on the sphere with $O(g)$ layers.

Unfortunately, the discrete theory is more rigid than the continuous one, and this will turn out to be impossible. Instead, we shall actually pass to the continuous theory to prove our result. To do this, we shall provide a subdivision lemma that shows that it suffices to prove Theorem 8.1.3 for graphs that have circle packings with very small circles. We shall then show that the smooth map that we have from the Riemann surface to the sphere will take *almost* all of the circles of our circle packing to curves on the sphere that are *almost* circles. We will then show that this representation of our graph as an approximate circle packing is enough to provide our desired bounds.

# Chapter 10

# Introduction to Circle Packings

Our proof of Theorem 8.1.3 operates by obtaining a nice geometric realization of $G$. We obtain this realization using the theory of circle packings. In this section, we shall review the basics of circle packing theory and quote the main results that our proof will employ. For a more comprehensive treatment of this theory and a historical account of its origins, see [48].

Loosely speaking, a circle packing is a collection of circles on a surface with a given pattern of tangencies. We remark at the outset that the theory that we are discussing *is not the same* as the classical theory of sphere packing. Our theory is concerned with the combinatorics of the tangency patterns, *not* with the maximum number of circles that one can fit in a small region. The coincidence of nomenclature is just an unfortunate historical accident.

## 10.1 Planar Circle Packings

For simplicity, we begin by discussing circle packings in the plane.

**Definition 10.1.1.** A *planar circle packing* $\mathcal{P}$ is a finite collection of (possibly overlapping) circles $C_1, \ldots, C_n$ of respective radii $r_1, \ldots, r_n$ in the complex plane $\mathbb{C}$. If all of the $C_i$ have disjoint interiors, we say that $\mathcal{P}$ is *univalent*.

The *associated graph* $A(\mathcal{P})$ of $\mathcal{P}$ is the graph obtained by assigning a vertex $v_i$ to each circle $C_i$ and connecting $v_i$ and $v_j$ by an edge if and only if $C_i$ and $C_j$ are mutually tangent.

This is illustrated in Figures 10-1 and 10-2.

We thus associate a graph to every circle packing. It is clear that every graph associated to a univalent planar circle packing is planar. A natural question to ask is whether every planar graph can be realized as the associated graph of some planar

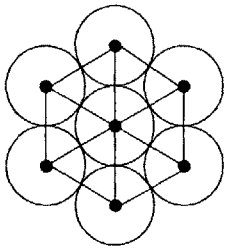

Figure 10-1: A univalent circle packing with its associated graph.

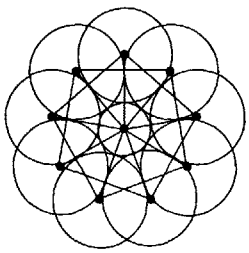

Figure 10-2: A nonunivalent circle packing with its associated graph.

circle packing. This is answered in the affirmative by the Koebe-Andreev-Thurston Theorem:

**Theorem 10.1.2** (Koebe-Andreev-Thurston). *Let $G$ be a planar graph. There exists a planar circle packing $\mathcal{P}$ such that $A(\mathcal{P}) = G$.*

This theorem also contains a uniqueness result, but we have not yet developed the machinery to state it. We shall generalize this theorem in Section 10.3, at which point we shall have the proper terminology to state the uniqueness part of the theorem.

We note that if we map the plane onto the sphere by stereographic projection, circles in the plane will be sent to circles on the sphere, so this theorem can be interpreted as saying that every genus 0 graph can be represented as a circle packing on the surface of a genus 0 surface. This suggests that we attempt to generalize this theorem to surfaces of higher genus. The theory of circle packings on surfaces of arbitrary genus acts in many ways like a discrete analogue of classical Riemann surface theory. As such, a basic background in Riemann surfaces is necessary to state or motivate many of its results. It is to this that we devote the next section.

## 10.2 A Very Brief Introduction to Riemann Surface Theory

In this section, we provide an informal introduction to Riemann surface theory. Our goal is to provide geometric intuition, not mathematical rigor. We assume some familiarity with the basic concept of a manifold, as well as with the basic definitions of complex analysis. For a more complete exposition of the theory, see [21].

We recall that an *n-dimensional manifold* is a structure that looks locally like $\mathbb{R}^n$. More formally, we write our manifold $M$ as a topological union of open sets $S_i$, each endowed with a homeomorphism $\varphi_i : S_i \to B_n$, where $B_n$ is the ball $\{|x| < 1 \mid x \in \mathbb{R}^n\}$. Furthermore, we require a compatibility among these maps to avoid cusps and such. To this end, we mandate that the compositions $\varphi_j \circ \varphi_i^{-1} : \varphi_i(S_i \cap S_j) \to \varphi_j(S_i \cap S_j)$ be diffeomorphisms. The orientable 2-dimensional manifolds are precisely the genus $g$ surfaces described above.

An *n-dimensional complex manifold* is the natural complex analytic generalization of this. We write our manifold $M$ as a union of open sets $S_i$ and endow each such set with a homeomorphism $\varphi_i : S_i \to B_{\mathbb{C}^n}$, where $B_{\mathbb{C}^n}$ is the complex unit ball $\{|x| < 1 \mid x \in \mathbb{C}^n\}$. Now, instead of requiring the compositions of these functions to obey a smooth compatibility condition, we require them to obey an analytic one: we demand that the compositions $\varphi_i \circ \varphi_j^{-1}$ be biholomorphic maps.

As such, an $n$-dimensional complex manifold $M$ is a $2n$-dimensional real manifold with additional complex analytic structure. This structure allows us to transfer over many of the definitions from standard complex analysis. The basic idea is that we define these notions as before on the $S_i$, and the compatibility condition allows them to make sense as global definitions. In particular, if $M = (S_i^M, \phi_i^M)$ and $N = (S_j^N, \phi_j^n)$ are complex manifolds of the same dimension, we say that a function $f : M \to N$ is holomorphic if its restriction to a map $f_{ij} : S_i^M \to S_j^N$ is holomorphic for all $i$ and $j$. Since the compositions $\varphi_i^M \circ (\varphi_j^M)^{-1}$ and $\varphi_i^N \circ (\varphi_j^N)^{-1}$ are holomorphic, this notion makes sense where the regions overlap.

**Definition 10.2.1.** A *Riemann surface* is a one-dimensional complex manifold.

In this paper, we shall take all of our Riemann surfaces to be compact. Since there is a natural way to orient the complex plane, we note that the complex structure can be used to define an orientation on the manifold. As such, all complex manifolds, and, in particular, all Riemann surfaces, are orientable. Compact Riemann surfaces are thus, topologically, two-dimensional orientable real manifolds. Every compact Riemann surface is therefore topologically one of the genus $g$ surfaces discussed above. The complex structure imposed by the $\varphi_i$, however, varies much more widely, and there are many different such structures that have the same underlying topological space.

Nothing in the definition of a Riemann surface supplies a metric on the surface. Indeed, there is no requirement that the different $\phi_i$ agree in any way about the distance between two points in their intersection. One can assign many different metrics to the surface. However, it turns out that there is way to single out a unique metric on the surface, called the metric of constant curvature. This allows us to supply an intrinsic notion of distance on any Riemann surface. In particular, this allows us to define a circle on our Riemann surface to be a simple closed curve that is contractible on the surface and all of whose points lie at a fixed distance from some center.

One particulary important Riemann surface that we shall consider is the Riemann sphere, which we denote $\widehat{\mathbb{C}}$. It is topologically a sphere. It should be thought of as being obtained by taking the complex plane and adjoining a single point called $\infty$. One way of visualizing its relation to $\mathbb{C}$ is to consider the stereographic projection away from the North Pole of a sphere, onto a plane. The North Pole corresponds to $\infty$, and the rest of the sphere corresponds to $\mathbb{C}$.

We recall from single variable complex analysis that the requirement that a map be analytic is quite a stringent one, and that it imposes a significant amount of local structure on the map. Let $f : \mathbb{C} \to \mathbb{C}$ be nonconstant and analytic in a neighborhood of the origin, and assume without loss of generality that $f(0) = 0$. There is some neighborhood of the origin in which $f$ can be expressed as a power series $f(z) = a_1 z + a_2 z^2 + a_3 z^3 + \ldots$. If $a_1 \neq 0$, $f(z)$ is analytically invertible in some neigbhorhood of the origin, so it is locally an isomorphism. In particular, it is *conformal*—it preserves the angles between intersecting curves, and the image of an infinitesimal circle is another infinitesimal circle.

If $a_1 = 0$ and $a_n$ is the first nonzero coefficient in its power series, $f$ has a *branch point of order* $n$ at the origin. In this case, $f$ operates, up to a scale factor and lower order terms, like the function $f(z) = z^n$. This function is $n$-to-1 on a small neighborhood of the origin, excluding the origin itself. It sends only 0 to 0, however. The preimages of the points in this small neighborhood thus trace out $n$ different "sheets" that all intersect at 0. This confluence of sheets is the only sort of singularity that can appear in an analytic map. We note that the angles between curves intersecting at the branch point are not preserved, but they are instead divided by $n$.

This local behavior is identical for Riemann surfaces. From this, we can deduce that if $f : M \to N$ is an analytic map of Riemann surfaces, it has some well-defined degree $k$. For all but finitely many points $p$ in $N$, $\#f^{-1}(p) = k$. The preimage of each of these points looks like a collection of $k$ sheets, and $f$ has nonzero derivative at all of them. There exist some points $q \in M$ at which $f'(q) = 0$. At each such point there is a branch point, so the sheets intersect, and $f(q)$ has fewer than $k$ preimages.

However, the global structure of Riemann surfaces provides further constraints on

maps between them, and there are, generally speaking, very few functions $f : M \to N$ of a given degree. For example, topological arguments, using the local form of analytic maps described above, show that there are no degree 1 maps from the torus to the sphere, and no degree 2 maps from the genus 2 surface to the sphere.

There is a deep theory of maps of Riemann surfaces that describes rather precisely when a map of a given degree exists between two Riemann surfaces, and, if it exists, where and how such a map must branch. Of this theory we shall only require one main result, which is a direct corollary of the celebrated Riemann-Roch theorem:

**Theorem 10.2.2.** *Let $M$ be a Riemann surface of genus $g$. There exists an analytic map $f : M \to \widehat{\mathbb{C}}$ of degree $O(g)$ and with $O(g)$ branch points.*

## 10.3 Circle Packings on Surfaces of Arbitrary Genus

We now have the machinery in place to deal with general circle packings. Throughout this section, let $G$ be a graph of genus $g$, and suppose that it is embedded on a genus $g$ surface $S$ so that none of its edges cross. The graph $G$ divides $S$ into faces. We say that $G$ is a *fully triangulated* graph if all of these faces are triangles, in which case we say that it gives a *triangulation* of $S$. If $G$ is not fully triangulated, one can clearly add edges to it to make it so. It will follow immediately from equation (11.2) in Chapter 11 that this will only increase $\lambda_2(G)$, so we shall assume for convenience that $G$ gives a triangulation of $S$. We are now ready to define our primary objects of study:

**Definition 10.3.1.** Let $S$ be a compact Riemann surface endowed with its metric of constant curvature. A *circle packing $\mathcal{P}$ on $S$* is a finite collection of (possibly overlapping) circles $C_1, \ldots, C_n$ of respective radii $r_1, \ldots, r_n$ on the surface of $S$. If all of the $C_i$ have disjoint interiors, we say that $\mathcal{P}$ is *univalent*.

The *associated graph* $A(\mathcal{P})$ of $\mathcal{P}$ is the graph obtained by assigning a vertex $v_i$ to each circle $C_i$ and connecting $v_i$ and $v_j$ by an edge if and only if $C_i$ and $C_j$ are mutually tangent. Alternatively, we say that $\mathcal{P}$ is a circle packing for $A(\mathcal{P})$ on $S$.

The main result on circle packings that we shall use is the Circle Packing Theorem, which is the natural extension of the Koebe-Andreev-Thurston Theorem to this more general setting. It was originally proven in a restricted form by Beardon and Stephenson [3] and then proven in full generality by He and Schramm [29].

**Theorem 10.3.2** (Circle Packing Theorem)**.** *Let $G$ be a triangulation of a surface of genus $g$. There exists a Riemann surface $S$ of genus $g$ and a univalent circle packing $\mathcal{P}$ such that $\mathcal{P}$ is a circle packing for $G$ on $S$. This packing is unique up to automorphisms of $S$.*

If $G$ is embedded in a surface of genus $g$ but is not fully triangulated, the Riemann surface and circle packing guaranteed by the theorem still exist, but they need not be unique.

The complex structure on the Riemann surface allows us to define the angle at which two edges of a face meet. If the points $u$, $v$, and $w$ are the vertices of a face, we denote the angle between the edges $\overline{uv}$ and $\overline{vw}$ at $v$ by $\langle uvw \rangle$. We can thus define the angle sum at a vertex to be $\sum \langle uvw \rangle$, where the sum is taken over all faces containing $v$. If $\mathcal{P}$ is a univalent sphere packing, the angle sum at any vertex of $A(\mathcal{P})$ is clearly $2\pi$.

In a nonunivalent circle packing, it is possible for the circles at a point to wrap around the point more than once. In the case of a nonunivalent circle packing, the edges of its associated graph may intersect, but we can still define an associated triangulation of the surface—there just may be more than one triangle covering a given point. We can therefore compute the angle sum at a point. In this case, it need not be $2\pi$. However, the circles must wrap around the vertex an integral number of times, so it must be some multiple $2\pi k$. (See Figure 10-2.) We then say that the vertex is a *discrete branch point of order k*.

These discrete branch points behave very much like the continuous branch points present on Riemann surfaces. In fact, there is an extensive theory that shows that a large portion of the theory of Riemann surfaces has an analogue in the discrete realm of circle packing. One can define maps of circle packings, just as one can define maps of Riemann surfaces. They consist of a correspondence of the circles on one surface to those on another in a way that commutes with tangency. While analytic maps send infinitesimal circles to infinitesimal circles, maps of circle packings send finite circles to finite circles. The analogue of branched covering maps in Riemannian geometry takes univalent circle packings and places them as nonunivalent circle packings on other surfaces. Unfortunately, these maps are somewhat rarer than their continuous analogues.

In particular, if we have a circle packing on a genus $g$ surface $S$, there is no known analogue of the Riemann-Roch theorem, and thus no analogue of Theorem 10.2.2. We are therefore not guaranteed that there is a nonunivalent circle packing on the sphere carrying the same associated graph. Intuitively, this comes from the fact that the analytic maps from $S$ to $\widehat{\mathbb{C}}$ are required to be branched over a very restricted locus of points. The discrete maps, however, can only be branched over the centers of circles. If there does not exist an admissible set of branch points among the centers of the circles, we will have difficulty constructing a discrete analytic map. This will lie at the root of many of the technical difficulties that we shall face in the remainder of this paper.

# Chapter 11

# An Eigenvalue Bound

In this section, we prove Theorem 8.1.3. The proof will assume a technical lemma whose proof we shall postpone until Chapter 12.

We begin by recalling the expression of the Fiedler value of $G$ as a so-called *Rayleigh quotient*:

$$\lambda_2 = \min_{x \perp (1,\ldots,1)^T} \frac{x^T L(G) x}{x^T x}. \tag{11.1}$$

A straightforward calculation shows that for $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$,

$$x^T L(G) x = \sum_{(i,j) \in E} (x_i - x_j)^2,$$

so that equation (11.1) becomes

$$\lambda_2 = \min_{x \perp (1,\ldots,1)^T} \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{x^T x}. \tag{11.2}$$

As noted by Spielman and Teng [47], it follows easily from equation (11.2) that we can replace the scalar values $x_i$ with vectors $v_i \in \mathbb{R}^k$, so that

$$\lambda_2 = \min \frac{\sum_{(i,j) \in E} \|v_i - v_j\|^2}{\sum_{i=1}^n \|v_i\|^2}, \tag{11.3}$$

where the minimum is taken over all sets of $n$-vectors such that $\sum v_i = (0, \ldots, 0)^T$ and such that at least one of the $v_i$ is nonzero.

The general goal is thus to find a set of $v_i$ that gives a small value for this quotient. The $v_i$ that we use will almost be the centers of a nonunivalent circle packing on the unit sphere $S^2 \subseteq \mathbb{R}^3$. The efficacy of this follows from the following theorem, which

follows easily from the work of Spielman and Teng [47].

**Theorem 11.0.3.** *Let $\mathcal{P}$ be a circle packing on the sphere $S^2 = \{x \in \mathbb{R}^3 \mid \|x\|^2 = 1\}$ so that the graph $A(\mathcal{P})$ has no vertex of degree greater than $\Delta$. Suppose further that the packing is of degree $k$, so that no point on the sphere is contained in the interior of more than $k$ circles, and that the centroid of the centers of the circles is the origin. Then the Fiedler value*

$$\lambda_2(A(\mathcal{P})) \leq O(\Delta k/n).$$

*Proof.* This follows from equation (11.3). Let the circles be $C_1, \ldots, C_n$, and let the corresponding radii be $r_1, \ldots, r_n$. Let $v_i \in \mathbb{R}^3$ be the $x$, $y$, and $z$ coordinates of the center of the $i^{\text{th}}$ circle. The sum $\sum v_i = 0$ by assumption, so $\lambda_2$ is less than or equal to the fraction in equation (11.3). Since all of the $v_i$ are on the unit sphere, we have $\sum \|v_i\|^2 = n$, so it just remains to bound the numerator. If there is an edge $(i, j)$, the two circles $C_i$ and $C_j$ must be mutually tangent, so that $\|v_i - v_j\|^2 \leq (r_i + r_j)^2 \leq 2(r_i^2 + r_j^2)$. It thus follows that

$$\sum_{(i,j)\in E} \|v_i - v_j\|^2 \leq \sum_{(i,j)\in E} 2(r_i^2 + r_j^2) \leq 2\Delta \sum_{i=1}^{n} r_i^2.$$

However, the total area of all of the circles is less than or equal to $k$ times the area of the sphere, since the circle packing is of degree $k$. We thus have that $\sum_{i=1}^{n} r_i^2 \leq O(k)$, from which the desired result follows. □

This suggests that we use the Circle Packing Theorem (Theorem 10.3.2) to embed our graph on a genus $g$ surface and then try to use some analogue of Theorem 10.2.2 to obtain a branched circle packing on the sphere of degree $O(g)$. Unfortunately, as previously noted, such a circle packing need not exist, due to the restrictiveness of the discrete theory. As such, we shall instead show that a certain subdivision process on our graph does not significantly decrease $n\lambda_2$. We shall then show that performing this subdivision enough times causes our discrete circle packing to approximate a continuous structure on the Riemann surface, at which point we can use the continuous theory in addition to the discrete one.

The refinement procedure that we shall use is called "hexagonal refinement." It operates on a triangulation of a surface by replacing each triangle with four smaller triangles, as shown in Figure 11-1. This process produces another triangulation of the same surface, so we can iterate it arbitrarily many times.

**Lemma 11.0.4** (Subdivison Lemma)**.** *Let $G$ be a graph with $n$ vertices, $m$ edges, and maximum degree $\Delta$ that triangulates some surface without boundary, and let $G'$ be*

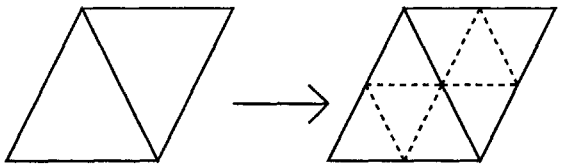

Figure 11-1: The hexagonal subdivision procedure applied to a triangulation with two triangles.

*the graph with $n'$ vertices and $m'$ edges obtained by performing $k$ successive hexagonal refinements on $G$. Then*

$$n\lambda_2(G) \leq C(\Delta)n'\lambda_2(G').$$

*Proof.* For the sake of continuity, we defer this proof to Chapter 12. □

The refinement process replaces each triangle in our graph with four smaller triangles. If all of the original triangles remained the same size and shape, this would imply that performing enough hexagonal refinements would give rise to a circle packing whose circles have arbitrarily small radii. However, it is possible for the original triangles to change size and shape as we refine, so this is no longer obvious. Nevertheless, it remains true, as shown by the following lemma:

**Lemma 11.0.5.** *Let $G$ be a graph that triangulates a genus $g$ Riemann surface without boundary, and let $G^{(k)}$ be the graph obtained by performing $k$ hexagonal refinements on $G$. For every $\epsilon > 0$, there exists some $k_\epsilon$ so that for all $\ell \geq k_\epsilon$, every circle in $G^{(\ell)}$ has radius less than $\epsilon$.*

*Proof.* This was essentially proven by Rodin and Sullivan [44]. Their proof, however, was only stated for the genus 0 case. The precise statement above was proven by Bowers and Stephenson [7]. □

We get a new Riemann surface for each iteration of the refinement procedure. It is intuitive that, as the number of iterations grows and the circles in the refined graph get arbitrarily small, the Riemann surfaces will somehow converge, and the embedding of the graph on these Riemann surfaces will somehow stabilize. This can be made formal by the following lemma:

**Lemma 11.0.6.** *Let $G$ be a graph that triangulates a genus $g$ compact Riemann surface without boundary, let $G^{(k)}$ be the result of performing $k$ hexagonal refinements on $G$, and let $S^{(k)}$ be the Riemann surface on which $G^{(k)}$ is realized as a circle packing. Further, let $h_k : S^{(k)} \to S^{(k+1)}$ be the map that takes a triangle to its image under the subdivision procedure by the obvious piecewise-linear map. The sequence of surfaces $\{S^{(k)}\}$ converges in the moduli space of genus $g$ surfaces, and the sequence of maps $\{h_k\}$ converges to the identity.*

*Proof.* This is proven by Bowers and Stephenson [7]. □

We shall also require one last definition:

**Definition 11.0.7.** Let $f : X \to Y$ be a map between two locally Euclidean metric spaces. The quantity

$$H_f(x, r) = \frac{\max_{|x-y|=r} |f(x) - f(y)|}{\min_{|x-y|=r} |f(x) - f(y)|} - 1.$$

is called the *radius $r$ distortion of $f$ at $x$.*

We are now finally ready to prove Theorem 8.1.3.

*Proof of Theorem 8.1.3.* Using the Circle Packing Theorem (Theorem 10.3.2), realize the graph $G = G^{(0)}$ as a circle packing on some Riemann surface $S$ of genus $g$. Let $G^{(k)}$ be the result of performing $k$ hexagonal refinements on $G$, and let $S^{(k)}$ be the Riemann surface on which it can be realized as a circle packing. By Theorem 10.2.2, there exists an analytic map $f^{(k)}$ from $S^{(k)}$ to the Riemann sphere of degree $O(g)$ and with $O(g)$ branch points. Embed the Riemann sphere as the unit sphere in $\mathbb{R}^3$ using the conformal map given by inverse stereographic projection. By the work of Spielman and Teng (Theorem 9 of [47]), post-composing with a Möbius transformation allows us to assume, without loss of generality, that the centroid of the images of the vertices of each $G^{(k)}$ under $f^{(k)}$ is the origin. By Lemma 11.0.6, the $S^{(k)}$ converge to some surface $S^{(\infty)}$, and the $f^{(k)}$ can be chosen so as to converge to some continuous limit map $f^{(\infty)}$.

By Lemma 11.0.4, it suffices to the prove the theorem for an arbitrarily fine hexagonal refinement of the original graph. Away from its branch points, a map of Riemann surfaces is conformal, meaning it sends infinitesimal circles to infinitesimal circles. In particular, given a map $f : S \to \widehat{\mathbb{C}}$, the compactness of $S$ guarantees that for every $\epsilon, \kappa > 0$, there exists a $\delta > 0$ so that the radius $\delta'$ distortion $H_f(x, \delta')$ is less than $\epsilon$ for every $x$ that is at least distance $\kappa$ from any branch point and any $\delta' \leq \delta$. In fact, by the convergence results of the last paragraph, there exist some $N$ and $\delta$ such that this holds for every $f^{(k)}$ with $k > N$. Fix $\epsilon$ and $\kappa$, and let $\delta$ and $N$ be chosen so that this is true. By possibly increasing $N$ if necessary, we can assume by Lemma 11.0.5 that all of the circles on $S^{(k)}$ have radius at most $\delta$ for all $k > N$.

Let $k$ be at least $N$. We shall break $S^{(k)}$ into two parts, $S^{(k)} = S_1^{(k)} \cup S_2^{(k)}$, as follows. Construct a ball of radius $\kappa$ around each branch point of $f^{(k)}$, and let $S_2^{(k)}$ be the union of these balls. Let $S_1^{(k)}$ be the complement $S^{(k)} - S_2^{(k)}$.

We can now use equation (11.3) to bound $\lambda_2$, just as in the proof of Theorem 11.0.3. Let $G^{(k)}$ have $n_k$ vertices. The denominator of equation (11.3) is equal to $n_k$, so it

suffices to bound the numerator. We shall consider separately the circles contained entirely in $S_1^{(k)}$ and those that intersect $S_2^{(k)}$.

We begin with the circles contained in $S_1^{(k)}$. Every circle of the packing gets mapped by $f$ to some connected region on $\widehat{\mathbb{C}}$, and there are at most $O(g)$ such regions covering any point of the sphere. Let $C$ be a circle in $S_1^{(k)}$, let $D$ be the diameter function, which takes a region to the length of the longest geodesic it contains, and let $A$ be the area function. Since the radius $\delta$ distortion of $f$ inside of $S_1^{(k)}$ is at most $\epsilon$, and the radius of $C$ is at most $\delta$, the ratio $D^2(f(C))/A(f(C))$ is at most $O(1+\epsilon)$. Using the same argument as in the proof of Theorem 11.0.3, the vertex at the center of a circle $C$ cannot contribute more than $O(dD^2(f(C)))$ to the sum, and the total area of the regions from $S_1^{(k)}$ cannot exceed $O(g)$, so the total contribution to the numerator of the vertices in $S_1^{(k)}$ cannot be more than $O(dg(1+\epsilon))$.

If this were the only term in the numerator, we could complete the proof by setting $\epsilon$ to be a constant. It thus remains to show that the contribution from the circles intersecting $S_2^{(k)}$ can be made small. To do this, we need only show that the contribution $\theta^{(k)}(x)$ to the numerator per unit area at a point $x$ from these circles remains bounded as we subdivide, since we can make the area of $S_2^{(k)}$ arbitrarily small by sending $\kappa$ to zero, and thus the area of the circles intersecting $S_2^{(k)}$ will go to zero as $k$ goes to infinity and the circles get arbitrarily small.

Let $x_i$, $i = 1, \ldots, 3$, be the coordinate functions on $\mathbb{R}^3$, and let $f^{(k)*}x_i$ be their pullbacks along $f^{(k)}$ to $S^{(k)}$. (That is, if $y$ is a point on $S^{(k)}$, $f^{(k)*}x_i(y) = x_i(f^{(k)}(y))$.) In addition, let $C_1^{(k)}$ and $C_2^{(k)}$ be a pair of adjacent circles in $S_2^{(k)}$ with respective radii $r_1^{(k)}$ and $r_2^{(k)}$ and respective centers $c_1^{(k)}$ and $c_2^{(k)}$. The contribution of the corresponding edge in $G^{(k)}$ to the numerator of equation (11.3) will be

$$\left\| \left( f^{(k)*}x_i(c_1^{(k)}) \right)_{i=1}^{3} - \left( f^{(k)*}x_i(c_2^{(k)}) \right)_{i=1}^{3} \right\|^2 \qquad (11.4)$$
$$= \sum_{i=1}^{3} \left( f^{(k)*}x_i(c_1^{(k)}) - f^{(k)*}x_i(c_2^{(k)}) \right)^2 .$$

The distance between $c_1^{(k)}$ and $c_2^{(k)}$ equals $r_1^{(k)} + r_2^{(k)}$. As $k$ goes to infinity, the radii $r_1^{(k)}$ and $r_2^{(k)}$ both go to zero, by Lemma 11.0.5. By the smoothness of the $f^{(k)}$, their convergence to $f^{(\infty)}$, and the compactness of their domains, we can approximate each term on the right-hand side of equation (11.4) arbitrarily well by its first order

approximation, so that

$$\left(f^{(k)*}x_i(c_1^{(k)}) - f^{(k)*}x_i(c_2^{(k)})\right)^2 \tag{11.5}$$
$$\leq (1+o(1))(r_1^{(k)}+r_2^{(k)})^2\|\nabla f^{(k)*}x_i(c_1^{(k)})\|^2$$

as $k$ goes to infinity and the distance between $c_1^{(k)}$ and $c_2^{(k)}$ shrinks to zero.

The right-hand side of equation (11.5) is bounded above by

$$(2+o(1))[(r_1^{(k)})^2+(r_2^{(k)})^2]\|\nabla f^{(k)*}x_i(c_1^{(k)})\|^2 \tag{11.6}$$
$$= O(1)[(r_1^{(k)})^2\|\nabla f^{(k)*}x_i(c_1^{(k)})\|^2 + (r_2^{(k)})^2\|\nabla f^{(k)*}x_i(c_2^{(k)})\|^2].$$

The degree of our graph is bounded, so every vertex appears in at most a constant number of edges. If we sum the right-hand side of equation (11.6) over all of the edges in our graph, the total contribution of terms involving a fixed circle of radius $r$ centered at $c$ is thus bounded above by

$$O(1)r^2\|\nabla f^{(k)*}x_i(c)\|^2,$$

so the contribution per unit area is bounded above by

$$O(1)\|\nabla f^{(k)*}x_i(c)\|^2.$$

This clearly remains bounded as $k$ goes to infinity and $f^{(k)}$ approaches $f^{(\infty)}$. It thus follows that the contribution to the numerator of equation (11.3) of the vertices in $S_2^{(k)}$ tends to zero as $k$ goes to infinity and $\kappa$ is made arbitrarily small. By setting $\epsilon$ to be a constant and sending $\kappa$ to zero, Theorem 8.1.3 follows. □

# Chapter 12

# The Proof of the Subdivision Lemma

In this section, we shall prove Lemma 11.0.4. In proving this bound, it will be convenient to consider a weighted form of the Laplacian:

**Definition 12.0.8.** The *weighted Laplacian* $\mathcal{L}^W(G)$ of a graph $G$ is the matrix

$$\mathcal{L}^W(G) = W^{-1/2} L(G) W^{-1/2},$$

where $L(G)$ is the Laplacian of $G$, and $W$ is a diagonal matrix whose $i^{\text{th}}$ diagonal entry $w_i$ is strictly positive for all $i$.

We shall denote the eigenvalues of $\mathcal{L}^W(G)$ by $\widetilde{\lambda}_1^W(G) \leq \cdots \leq \widetilde{\lambda}_n^W(G)$ and the corresponding eigenvectors by $\widetilde{v}_1^W(G) \ldots \widetilde{v}_n^W(G)$. A straightforward calculation shows that the weighted Laplacian has $\widetilde{\lambda}_1^W = 0$ and $\widetilde{v}_1^W = W^{1/2}\mathbf{1}$. Our main quantity of interest will be $\widetilde{\lambda}_2^W(G)$, which we can compute using a weighted analogue of the Rayleigh quotient:

$$\widetilde{\lambda}_2^W = \min_{x \perp W\mathbf{1}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_i x_i^2 w_i}. \tag{12.1}$$

The second eigenvector $\widetilde{v}_2^W(G)$ equals $W^{1/2}x$, where $x$ is the vector that achieves the minimum in equation (12.1).

If all of the weights are $\Theta(1)$, standard linear algebra shows that $\lambda_2(G)$ and $\widetilde{\lambda}_2^W(G)$ differ by at most a constant factor, so proving a bound on one implies a bound on the other. (See Chung's book [12] for detailed proofs of the above facts and for other foundational information about the weighted Laplacian.)

Before we can proceed to the body of the proof of Lemma 11.0.4, we shall require two fairly general technical lemmas about independent random variables.

**Lemma 12.0.9.** *Let $a_1, \ldots, a_n$ be independent real-valued random variables, possibly drawn from different probability distributions. Let $w_1, \ldots, w_n \in \mathbb{R}^+$ be strictly positive constants. If the expectation $\mathbf{E}[\sum_i w_i a_i] = 0$, then*

$$\mathbf{E}\Big[\Big(\sum_j w_j a_j\Big)^2\Big] \leq \mathbf{E}\Big[\sum_j w_j^2 a_j^2\Big].$$

*Proof.* This follows by expanding out the left-hand side:

$$\begin{aligned}
\mathbf{E}\Big[\Big(\sum_j w_j a_j\Big)^2\Big] &= \mathbf{E}\Big[\sum_i w_i^2 a_i^2\Big] + \mathbf{E}\Big[\sum_i w_i a_i \Big(\sum_{j \neq i} w_j a_j\Big)\Big] \\
&= \mathbf{E}\Big[\sum_i w_i^2 a_i^2\Big] + \sum_i -\big(\mathbf{E}[w_i a_i]\big)^2 \\
&\leq \mathbf{E}\Big[\sum_j w_j^2 a_j^2\Big],
\end{aligned}$$

where the second equality follows from the independence of the variables and the fact that the sum of their expectations is zero. □

We shall now use this lemma to establish our second lemma, which is the one that will actually appear in our main proof:

**Lemma 12.0.10.** *Let $a_1, \ldots, a_n$ be independent real-valued random variables, possibly drawn from different probability distributions, and let $w_1, \ldots, w_n \in \mathbb{R}^+$ be strictly positive constants such that $\mathbf{E}[\sum_i w_i a_i] = 0$. Let $a = (a_1, \ldots, a_n)$, and let $w_{\max} = \max_i w_i$. Further let*

$$b = \left(\frac{1}{\sum_i w_i} \sum_i w_i a_i\right) \mathbf{1}, \quad \text{and let } c = a - b.$$

*Then*

$$\mathbf{E}\Big[\sum_i w_i c_i^2\Big] \geq \left(1 - \frac{w_{\max}}{\sum_i w_i}\right) \mathbf{E}\Big[\sum_i w_i a_i^2\Big].$$

*Proof.* This follows by direct calculation:

$$\begin{aligned}
\mathbf{E}\Big[\sum_i w_i c_i^2\Big] &= \mathbf{E}\Big[\sum_i w_i\Big(a_i - \frac{1}{\sum_j w_j}\big(\sum_j w_j a_j\big)\Big)^2\Big] \\
&= \mathbf{E}\Big[\sum_i w_i a_i^2\Big] + \frac{1}{\big(\sum_i w_i\big)^2}\mathbf{E}\Big[\sum_i w_i\big(\sum_j w_j a_j\big)^2\Big] - \frac{2}{\sum_i w_i}\mathbf{E}\Big[\sum_i w_i a_i\big(\sum_j w_j a_j\big)\Big] \\
&= \mathbf{E}\Big[\sum_i w_i a_i^2\Big] + \frac{1}{\sum_i w_i}\mathbf{E}\Big[\big(\sum_j w_j a_j\big)^2\Big] - \frac{2}{\sum_i w_i}\mathbf{E}\Big[\big(\sum_j w_j a_j\big)^2\Big] \\
&= \mathbf{E}\Big[\sum_i w_i a_i^2\Big] - \frac{1}{\sum_i w_i}\mathbf{E}\Big[\big(\sum_j w_j a_j\big)^2\Big] \\
&\geq \mathbf{E}\Big[\sum_i w_i a_i^2\Big] - \frac{1}{\sum_i w_i}\mathbf{E}\Big[\sum_j w_j^2 a_j^2\Big] \\
&= \mathbf{E}\Big[\sum_i \Big(1 - \frac{w_i}{\sum_j w_j}\Big) w_i a_i^2\Big] \\
&\geq \Big(1 - \frac{w_{\max}}{\sum_i w_i}\Big)\mathbf{E}\Big[\sum_i w_i a_i^2\Big],
\end{aligned}$$

where second-to-last inequality follows from Lemma 12.0.9. □

We are now prepared to prove Lemma 11.0.4.

*Proof of Lemma 11.0.4.* Let $G = (V_G, E_G)$ be the original graph, and let $G' = (V_{G'}, E_{G'})$ be the graph that results from performing $k$ successive hexagonal refinements on $G$. The embeddings into surfaces endow both $G$ and $G'$ with triangulations; let $T_G$ and $T_{G'}$ be the respective sets of triangles in these triangulations. There is a natural inclusion $\iota : V_G \hookrightarrow V_{G'}$, since the subdivision procedure only adds vertices to the original set. There is also a map $\eta : T_{G'} \to T_G$ that takes a triangle from the subdivided graph to the one in the original graph from which it arose. For a vertex $v$ in either graph, let $N(v)$ be the set of triangles containing it. For a vertex $w \in V_G$, let $P(w) = \eta^{-1}(N(w))$ be the set of triangles in $T(G')$ taken by $\eta$ to elements of $N(w)$. (See Figure 12-1.)

Our proof will proceed by producing a randomized construction of a subgraph $H$ of $G'$. Given a vector that assigns a value to every vertex of $G'$, we can obtain such a vector on $H$ by restriction. We shall also show how to use such a vector on $H$ to construct such a vector on $G$. The vectors on the different graphs will give rise to Rayleigh quotients on the graphs (some of which will be weighted), where the Rayleigh quotients for $G$ and $H$ will depend on the random choices made in the construction of $H$. By relating the terms in the different Rayleigh quotients, we shall

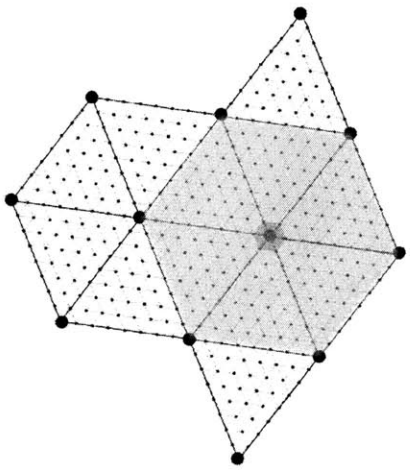

Figure 12-1: A subdivided graph, with $P(w)$ and $N(w)$ shaded for a vertex $w$.

then provide a probabilistic proof that there exists an $H$ that gives rise to a small Rayleigh quotient on $G$, which will suffice to prove our desired bound.

$H$ will be produced by randomly choosing a representative in $V_{G'}$ for each vertex in $V_G$ and representing every edge in $E_G$ by a randomly chosen path in $G'$ between the representatives of its endpoints.

We first construct the map $\pi_V : V_G \to V_{G'}$ that chooses the representatives of the vertices. For each $v \in V_G$ we choose $\pi_V(v)$ uniformly at random from the vertices contained in $P(v)$ that are at least as close to $\iota(v)$ as to $\iota(w)$ for any other $w \in V_G$. Vertices in $P(v)$ that are equally close to $\iota(v)$ and $\iota(w)$ should be arbitrarily assigned to either $v$ or $w$, but not both.

We now construct $\pi_E$, which maps edges in $E_G$ to paths in $G'$. Let $e = (v_1, v_2)$ be an edge in $G$, and let $w_1$ and $w_2$ equal $\pi_V(v_1)$ and $\pi_V(v_2)$ respectively. The two neighborhoods in $G$, $N(v_1)$ and $N(v_2)$, share exactly two triangles, $t_1$ and $t_2$. Let $x$ be a vertex randomly chosen from the vertices in $\eta^{-1}(t_1 \cup t_2)$. We shall construct a path from each $w_i$ $(i = 1, 2)$ to $x$, so that their composition gives a path from $w_1$ to $w_2$. We shall use the same construction for each, so, without loss of generality, we shall just construct the path from $w_1$ to $x$.

Both $w_1$ and $x$ are in $P(v_1)$, and we give a general procedure for constructing a path between any two such vertices. The images under the inclusion $\iota$ of the triangles in $N(v_1)$ encircle $\iota(v_1)$. Suppose $w_1$ is contained in $T_1$, and $x$ is contained in $T_2$. Traversing the triangles in a clockwise order from $T_1$ to $T_2$ gives one list of triangles,

and traversing in a counterclockwise order gives another. Let $T_1, Q_1, \ldots Q_\ell, T_2$ be the shorter of these two lists, with a random choice made if the two lists are the same length. Choose a random vertex $a_i$ in each $Q_i$, and let $a_0 = w_1$ and $a_{\ell+1} = x$. We thus have a vertex representing each triangle in the list. Our path will consist of a sequence of segments from each representative to the next.

Note that all of the triangles are distinct, except if $T_1 = T_2$ and the list is of length 2. We suppose for now that we have two vertices $a_i$ and $a_{i+1}$ in distinct triangles, and we deal with the degenerate case later. The two triangles in question are adjacent, and their union contains a grid graph as a subgraph. (See Figure 12-2.) Given two vertices in a grid, there is a unique path between them that one obtains by first moving horizontally and then vertically, and another that one obtains by moving vertically and then horizontally. (These two coincide if there is a line connecting the two points.) Randomly choose one of these two paths. This is the path connecting $a_i$ to $a_{i+1}$. If $a_i$ and $a_{i+1}$ lie in the same triangle, randomly choose one of the two adjacent triangles to form a grid, and then use the above construction. Composing the paths between each $a_i$ and $a_{i+1}$ completes the construction of $\pi_E$. The entire construction is illustrated in Figure 12-3.

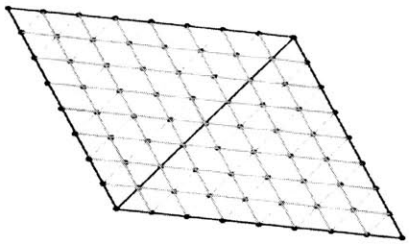

Figure 12-2: An illustration of how the grid graph exists as a subgraph of the union of two adjacent subdivided triangles.

We now consider the Rayleigh quotients for the three graphs that we have constructed. After $k$ hexagonal refinements, every edge in $G$ is split into $r = 2^k$ pieces, every triangle gets replaced with $r^2$ smaller triangles, and the number of vertices grows quadratically in $r$. A vector $y \in \mathbb{R}^{|V_{G'}|}$ that assigns a value to each vertex in $G'$ gives the Rayleigh quotient

$$R(G') = \frac{\sum_{(i,j)\in E_{G'}} (y_i - y_j)^2}{y^T y}.$$

This induces a vector on the vertices of $H$ by restriction. The probability, taken over the random choices in the construction of $\pi_V$ and $\pi_E$, that a given edge of $G'$

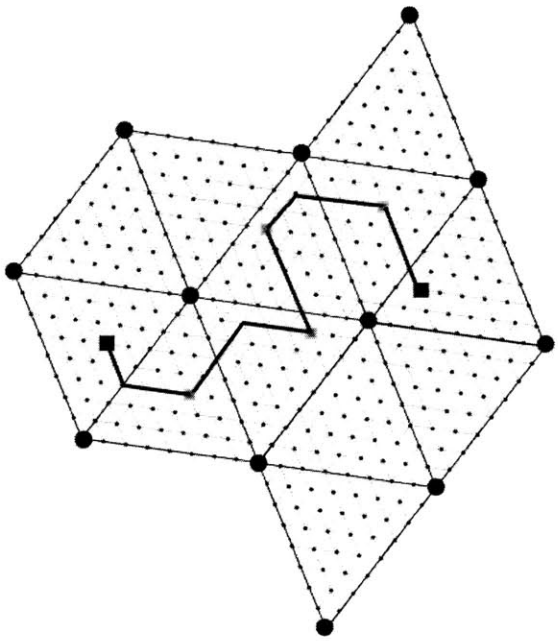

Figure 12-3: The entire construction illustrated for a given edge of the original graph.

appears on the path representing a given edge $e$ of $G$ is zero if it is not in $P(\alpha)$ with $\alpha$ equal to one of the endpoints of $e$, and at most $O(1/r)$ otherwise. Since the maximum degree of a vertex in $G$ is assumed constant, the expected number of times that a given edge of $G'$ occurs in $H$ is $O(1/r)$. Every vertex in $G'$ is selected as a representative of a vertex in $G$ with probability $\Theta(1/r^2)$. It thus follows that

$$\mathbf{E}\left[\sum_{(i,j)\in E_H}(y_i - y_j)^2\right] \leq O(1/r)\sum_{(i,j)\in E_{G'}}(y_i - y_j)^2, \tag{12.2}$$

and

$$\mathbf{E}\left[\sum_{i\in V_G} w_i y^2_{\pi_V(i)}\right] = \Theta(1/r^2)\sum_{i\in V_{G'}} y_i^2, \tag{12.3}$$

where the expectations are taken over the random choices in the construction of $(\pi_V, \pi_E)$, and the $w_i$ are any weights that are bounded above and below by positive constants.

Let $\overline{y}$ be the vector in $\mathbb{R}^{|V_G|}$ whose $i^{\text{th}}$ coordinate is $y_{\pi_V(i)}$. Each coordinate $\overline{y}_i$ of $\overline{y}$ is chosen independently from a distinct set $S_i$ of the coordinates of $y$, and every coordinate is contained in one of these sets. Let $s_i = |S_i|$, let $s_{\min} = \min_i s_i$, and take $W$ to be the diagonal matrix whose $i^{\text{th}}$ diagonal entry $w_i$ equals $s_i/s_{\min}$. The probability that a given vertex in $S_i$ is selected equals $1/s_i$, so we have that

$$\mathbf{E}\left[\sum_{j\in V_G} w_j \overline{y}_j\right] = \sum_{k\in V_{G'}} y_k = 0.$$

(The necessity to weight the terms on the left-hand side of this expression by the $w_i$ is what will necessitate the use of the weighted Laplacian in our proof.) The size of each $S_i$ is approximately proportional to the degree of the $i^{\text{th}}$ vertex of $G$, so the $w_i$ are all bounded above by a constant, and they are all at least one by definition. The eigenvalue $\widetilde{\lambda}_2^W(G)$ of the weighted Laplacian is thus within a constant factor of the standard Fiedler value $\lambda_2(G)$.

Let $z$ be the vector

$$z = \overline{y} - \left(\frac{\sum_i w_i \overline{y}_i}{\sum_i w_i}\right)\mathbf{1},$$

so that $z$ differs from $\overline{y}$ by a multiple of the all-ones vector and is orthogonal to $W\mathbf{1}$.

By applying Lemma 12.0.10 to equation (12.3), we obtain

$$\mathbf{E}\left[\sum_{i\in V_G} w_i z_i^2\right] \geq \left(1 - \frac{w_{\max}}{\sum_i w_i}\right)\mathbf{E}\left[\sum_{i\in V_G} w_i \overline{y}_i^2\right] = \Theta(1/r^2)\sum_{i\in V_{G'}} y_i^2. \tag{12.4}$$

Multiplying the inequalities in (12.2) and (12.4) by the appropriate factors and combining them yields

$$O(r)\left(\sum_{(i,j)\in E_{G'}} (y_i - y_j)^2\right)\cdot\mathbf{E}\left[\sum_{i\in V_G} w_i z_i^2\right] \geq \left(\sum_{i\in V_{G'}} y_i^2\right)\cdot\mathbf{E}\left[\sum_{(i,j)\in E_H} (y_i - y_j)^2\right] \tag{12.5}$$

This implies that there exists some choice of $(\pi_V, \pi_E)$ for which the left-hand side of (12.5) is greater than or equal to the right-hand side, in which case we would have

$$\begin{aligned}\frac{\sum_{(i,j)\in E_H}(y_i - y_j)^2}{\sum_{i\in V_G} w_i z_i^2} &\leq O(r)\frac{\sum_{(i,j)\in E_{G'}}(y_i - y_j)^2}{\sum_{i\in V_{G'}} y_i^2} \\ &= O(r)R(G'). \end{aligned}\tag{12.6}$$

Now suppose that we assign to each vertex $v \in V_G$ the value assumed by $y$ at $\pi_V(v)$. Using the fact that maximum degree of a vertex is bounded, so that there are $O(1)$ triangles surrounding any vertex in $G$, we see that every path representing an edge is of length $O(r)$. We note that if $i_1, \ldots, i_s$ is a sequence of vertices,

$$(y_{i_s} - y_{i_1})^2 \leq s\sum_{a=1}^{s-1}(y_{i_{a+1}} - y_{i_a})^2.$$

As such, we have

$$\sum_{(i,j)\in E_G} (y_{\pi_V(i)} - y_{\pi_V(j)})^2 \leq O(r)\sum_{(i,j)\in E_H} (y_i - y_j)^2. \tag{12.7}$$

Since $z$ is obtained from $\overline{y}$ by subtracting a multiple of the all-ones vector,

$$z_i - z_j = y_{\pi_V(i)} - y_{\pi_V(j)}$$

for any $i$ and $j$. Plugging this into equation (12.7) gives

$$\sum_{(i,j)\in E_G} (z_i - z_j)^2 \leq O(r)\sum_{(i,j)\in E_H} (y_i - y_j)^2,$$

and applying this to the inequality in (12.6) yields

$$\frac{\sum_{(i,j)\in E_G}(z_i - z_j)^2}{\sum_{i\in V_G} w_i z_i^2} \leq O(r^2)R(G').$$

We have thus constructed an assignment of values to the vertices of $G$ that is orthogonal to the vector $W\mathbf{1}$ and produces a weighted Rayleigh quotient of $O(r^2)R(G')$. If we choose the $y_i$ to be the values that give the Fiedler value of $G'$, we thus obtain, by equation (12.1) and the fact that the $w_i$ are $\Theta(1)$,

$$\lambda_2(G) = \Theta(1)\widetilde{\lambda}_2^W(G) \leq O(r^2)\lambda_2(G').$$

Since the number of vertices in $G'$ grows as $r^2$ times the number of vertices in $G$, this completes the proof of Lemma 11.0.4. □

# Bibliography


[1] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for non-planar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, October 1990.

[2] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: a survey. *Integration: the VLSI Journal*, 19:1–81, 1995.

[3] Alan F. Beardon and Kenneth Stephenson. The uniformization theorem for circle packings. *Indiana University Mathematics Journal*, 39:1383–1425, 1990.

[4] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[5] T. Bonnesen and W. Fenchel. *Theory of Convex Bodies*. B C S Associates, 1988.

[6] Karl Heinz Borgwardt. *The Simplex Method: a probabilistic analysis*. Number 1 in Algorithms and Combinatorics. Springer-Verlag, 1980.

[7] Philip Bowers and Kenneth Stephenson. Uniformizing dessins and Belyi maps via circle packing. To appear. Available at `http://web.math.fsu.edu/~bowers/Papers/recentPapers.html`, 2003.

[8] P.K. Chan, M. Schlag, and J. Zien. Spectral k-way ratio cut partitioning and clustering. In *Symposium on Integrated Systems*, 1993.

[9] T.F. Chan and D. C. Resasco. A framework for the analysis and construction of domain decomposition preconditioners. Technical report, UCLA, 1987.

[10] T.F. Chan and B. Smith. Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes. *Contemporary Mathematics*, pages 1–14, 1993.

[11] A. Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20:160–170, 1952.

[12] Fan R.K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.

[13] Vasek Chvatal. *Linear Programming.* W.H. Freeman, 1983.

[14] G. B. Dantzig. Maximization of linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. 1951.

[15] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[16] Amit Deshpande and Daniel A. Spielman. Improved smoothed analysis of the shadow vertex simplex method. preliminary version appeared in FOCS '05, 2005.

[17] H. Djidjev and J. Reif. An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs. In *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing*, pages 337–348, 1991.

[18] H.N. Djidjev. A linear algorithm for partitioning graphs. *Comptes rendus de l'Academie Bulgare des Sciences*, 35:1053–1056, 1982.

[19] John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 315–320, New York, June 13–15 2004. ACM Press.

[20] I.S. Filotti, G.L. Miller, and J.H. Reif. On determining the genus of a graph in $O(v^{O(g)})$ steps. In *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*, pages 27–37, 1979.

[21] Otto Forster. *Lectures on Riemann Surfaces.* Springer-Verlag, 1981.

[22] K. Fukuda and T. Terlaky. Criss-cross methods: A fresh view on pivot algorithms. *Mathematical Programming*, 79:369–395, 1997.

[23] Bernd Gartner, Martin Henk, and Gunter Ziegler. Randomized simplex algorithms on Klee-Minty cubes. *Combinatorica*, 18:349–372, 1998.

[24] S. Gass and Th. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.

[25] J.R. Gilbert, J. Hutchinson, and R. Tarjan. A separation theorem for graphs of bounded genus. *Journal of Algorithms*, 5:391–407, 1984.

[26] Martin Grotschel, Laszlo Lovasz, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1991.

[27] S. Guattery and G.L. Miller. On the performance of the spectral graph partitioning methods. In *Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 233–242, 1995.

[28] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on computer-aided design*, 11(9):1074–1085, September 1992.

[29] Zheng-Xu He and Oded Schramm. Fixed points, Koebe uniformization and circle packings. *Annals of Mathematics*, 137:369–406, 1993.

[30] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Ann. ACM Symp. on Theory of Computing*, pages 475–482, Victoria, B.C., Canada, May 1992.

[31] Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bulletin Amer. Math. Soc.*, 26:315–316, 1992.

[32] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[33] Jonathan A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. In *Symposium on the Theory of Computation (STOC)*, 2004.

[34] Jonathan A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. Master's thesis, Massachusetts Institute of Technology, 2005.

[35] Jonathan A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. *SIAM Journal on Computing, Special Issue for STOC 2004*, 35(4), 2006.

[36] Jonathan A. Kelner and Evdokia Nikolova. On the hardness and smoothed complexity of quasi-concave minimization. *To appear.*

[37] Jonathan A. Kelner and Daniel A. Spielman. A randomized polynomial-time simplex algorithm for linear programming. In *Symposium on the Theory of Computation (STOC)*, 2006.

[38] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademia Nauk SSSR*, pages 1093–1096, 1979.

[39] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal of Applied Mathematics*, 36:177–189, April 1979.

[40] N. Megiddo and R. Chandrasekaran. On the epsilon-perturbation method for avoiding degeneracy. *Operations Research Letters*, 8:305–308, 1989.

[41] G. Mihail. Conductance and convergence of markov chains—a combinatorial treatment of expanders. In *Proceedings of the 29th Annual IEEE Conference on Foundations of Computer Scienc*, pages 526–531, 1989.

[42] Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal of Discrete Mathematics*, 12:6–26, 1999.

[43] Evdokia Nikolova, Jonathan A. Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *Proceedings of the European Symposium on Algorithms*, 2006.

[44] Burt Rodin and Dennis Sullivan. The convergence of circle packings to the Riemann mapping. *J. Differential Geometry*, 26:349–360, 1987.

[45] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, 1986.

[46] H.D. Simon. Partitioning of unstructured problems for parallel processing. *Computer Systems in Engineering*, 2(2/3):135–148, 1991.

[47] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of the 37th Annual IEEE Conference on Foundations of Computer Science*, 1996.

[48] Ken Stephenson. Circle packing and discrete analytic function theory. Available online at `http://www.math.utk.edu/~kens`.

[49] C. Thomassen. The graph genus problem is NP-complete. *Journal of Algorithms*, 10(4):568–576, December 1989.

[50] Robert J. Vanderbei. *Linear Programming: Foundations and Extensions.* Springer, 2nd edition, 2001.

[51] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. Technical report, California Institute of Technology, 1990.