# Approximating the Largest Root and Applications to Interlacing Families

Nima Anari[1], Shayan Oveis Gharan[2], Amin Saberi[1], and Nikhil Srivastava[3]

[1]Stanford University, {anari,saberi}@stanford.edu
[2]University of Washington, shayan@cs.washington.edu
[3]University of California, Berkeley, nikhil@math.berkeley.edu

April 11, 2017

**Abstract**

We study the problem of approximating the largest root of a real-rooted polynomial of degree $n$ using its top $k$ coefficients and give nearly matching upper and lower bounds. We present algorithms with running time polynomial in $k$ that use the top $k$ coefficients to approximate the maximum root within a factor of $n^{1/k}$ and $1+O(\frac{\log n}{k})^2$ when $k \leq \log n$ and $k > \log n$ respectively. We also prove corresponding information-theoretic lower bounds of $n^{\Omega(1/k)}$ and $1+\Omega\left(\frac{\log \frac{2n}{k}}{k}\right)^2$, and show strong lower bounds for noisy version of the problem in which one is given access to approximate coefficients.

This problem has applications in the context of the method of interlacing families of polynomials, which was used for proving the existence of Ramanujan graphs of all degrees, the solution of the Kadison-Singer problem, and bounding the integrality gap of the asymmetric traveling salesman problem. All of these involve computing the maximum root of certain real-rooted polynomials for which the top few coefficients are accessible in subexponential time. Our results yield an algorithm with the running time of $2^{\tilde{O}(\sqrt[3]{n})}$ for all of them.

# Contents

# 1 Introduction

For a non-negative vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \in \mathbb{R}_+^n$, let $\chi_{\boldsymbol{\mu}}$ denote the unique monic polynomial with roots $\mu_1, \ldots, \mu_n$:

$$\chi_{\boldsymbol{\mu}}(x) := \prod_{i=1}^{n}(x - \mu_i).$$

Suppose that we do not know $\boldsymbol{\mu}$, but rather know the top $k$ coefficients of $\chi_{\boldsymbol{\mu}}$ where $1 \leq k < n$. In more concrete terms, suppose that

$$\chi_{\boldsymbol{\mu}}(x) = x^n + c_1 x^{n-1} + c_2 x^{n-2} + \cdots + c_n,$$

and we only know $c_1, \ldots, c_k$. What do $c_1, \ldots, c_k$ tell us about the roots and in particular $\max_i \mu_i$?

**Problem 1.1.** *Given the top $k$ coefficients of a real rooted polynomial of degree $n$, how well can you approximate its largest root?*

This problem may seem completely impossible if $k$ is significantly smaller than $n$. For example, consider the two polynomials $x^n - a$ and $x^n - b$. The largest roots of these two polynomials can differ arbitrarily in absolute value and even by knowing the top $n-1$ coefficients we can not approximate the absolute value of the largest at all.

The key to approach the above problem is to exploit real rootedness. One approach is to construct a polynomial with the given coefficients and study its roots. Unfortunately, even assuming that the roots of the original polynomial are real and well-separated, adding an exponentially small amount of noise to the (bottom) coefficients can lead to constant sized perturbations of the roots in the complex plane — the most famous example is Wilkinson's polynomial [Wil84]:

$$(x-1)(x-2)\ldots(x-20).$$

Instead, we use the given coefficients to compute a polynomial of the roots, e.g., the $k$-th moment of the roots, and we use that polynomial to estimate the largest root. Our contributions towards Problem 1.1 are as follows.

**Efficient Algorithm for Approximating the Largest Root.** In Theorem 3.1, we present an upper bound showing that we can use the top $k$ coefficients of a real rooted polynomial with nonnegative roots to efficiently obtain an $\alpha_{k,n}$ approximation of the largest root where

$$\alpha_{k,n} = \begin{cases} n^{1/k} & k \leq \log n, \\ 1 + O(\frac{\log n}{k})^2 & k > \log n. \end{cases}$$

Moreover such an approximation can be done in $\text{poly}(k)$ time. This implies that exact access to $O\left(\frac{\log n}{\sqrt{\epsilon}}\right)$ coefficients is sufficient for $(1+\epsilon)$-approximation of the largest root.

**Nearly Matching Lower Bounds.** The main nontrivial part of this work is our information-theoretic matching lower bounds. In Theorem 3.2, we show that when $k < n^{1-\epsilon}$ and for some constant $c$, there are no algorithms with approximation factor better than $\alpha_{k,n}^c$. Chebyshev polynomials are critical for this construction as well. We also use known constructions for large-girth graphs, and the proof of [MSS13a] for a variant of Bilu-Linial's conjecture [BL06]. Our bounds can be made slightly sharper assuming Erdős's girth conjecture.

## 1.1 Motivation and Applications

For many important polynomials, it is easy to compute the top $k$ coefficients exactly, whereas it is provably hard to compute all of them. One example is the matching polynomial of a graph, whose coefficients encode the number of matchings of various sizes. For this polynomial, computing the constant term, i.e. the number of perfect matchings, is #P-hard [Val79], whereas for small $k$, one can compute the number of matchings of size $k$ exactly, in time $n^{O(k)}$, by simply enumerating all possibilities. Roots of the matching polynomial, and in particular the largest root arise in a number of important applications [HL72, MSS13a, SS13]. So it is natural to ask how well the largest root can be approximated from the top few coefficients.

Another example of a polynomial whose top coefficients are easy to compute is the independence polynomial of a graph, which is real rooted for claw-free graphs [CS07], and whose roots have connections to the Lovász Local Lemma (see e.g. [HSV16]).

**Subexponential Time Algorithms for Method of Interlacing Polynomials** Our main motivation for this work is the method of interlacing families of polynomials [MSS13a, MSS13b, AO15b], which has been an essential tool in the development of several recent results including the construction of Ramanujan graphs via lifts [MSS13a, MSS15, HPS15], the solution of the Kadison-Singer problem [MSS13b], and improved integrality gaps for the asymmetric traveling salesman problem [AO15a]. Unfortunately, all these results on the existence of expanders, matrix pavings, and thin trees, have the drawback of being nonconstructive, in the sense that they do not give polynomial time algorithms for finding the desired objects (with the notable exception of [Coh16]). As such, the situation is somewhat similar to that of the Lovász Local Lemma (which is able to guarantee the existence of certain rare objects nonconstructively), before algorithmic proofs of it were found [Bec91, MT10].

In Section 4, we use Theorem 3.1 to give a $2^{\tilde{O}(\sqrt[3]{m})}$ time algorithm for rounding an interlacing family of depth $m$, improving on the previously known running time of $2^{O(m)}$. This leads to algorithms of the same running time for all of the problems mentioned above.

**Lower Bounds Given Approximate Coefficients** In the context of efficient algorithms, one might imagine that computing a much larger number of coefficients *approximately* might provide a better estimate of the largest root. In particular, we consider the following noisy version of Problem 1.1:

**Problem 1.2.** *Given real numbers $a_1, \dots, a_k$, promised to be $(1+\delta)-$approximations of the first $k$ coefficients of a real-rooted polynomial $p$, how well can you approximate the largest root of $p$?*

An important extension of our information theoretic lower bounds is that Problem 1.1 is extremely sensitive to noise: in Proposition 3.12 we prove that even knowing *all* but the $k$-th coefficient exactly and knowing the $k$-th one up to a $1+1/2^k$ error is no better than knowing only the first $k-1$ coefficients exactly. We do this by exhibiting two polynomials which agree on all their coefficients except for the $k^{th}$, in which they differ slightly, but nonetheless have very different largest roots.

This example is relevant in the context of interlacing families, because the polynomials in our lower bound have a common interlacing and are characteristic polynomials of $2-$lifts of a base graph, which means they could actually arise in the proofs of [MSS13a, MSS13b]. To appreciate this more broadly, one can consider the following taxonomy of increasingly structured polynomials:

$$\text{complex polynomials} \supset \text{real-rooted polynomials} \supset \text{mixed characteristic polynomials}$$
$$\supset \text{characteristic polynomials of lifts of graphs.}$$

Our example complements the standard numerical analysis wisdom (as in Wilkinson's example) that complex polynomial roots are in general terribly ill-conditioned as functions of their coefficients, and shows that this fact remains true even in the structured setting of interlacing families.

Proposition 3.12 is relevant to the quest for efficient algorithms for interlacing families for the following reason. All of the coefficients of the matching polynomial of a bipartite graph can be approximated to $1 + 1/\operatorname{poly}(n)$ error in polynomial time, for any fixed polynomial, using Markov Chain Monte Carlo techniques [SJ89, JSV04, FL06]. One might imagine that an extension of these techniques could be used to approximate the coefficients of the more general expected characteristic polynomials that appear in applications of interlacing families. In fact for some families of interlacing polynomials (namely, the mixed characteristic polynomials of [MSS13b]) we can design Markov chain Monte Carlo techniques to approximate the top half of the coefficients within $1 + 1/\operatorname{poly}(n)$ error.

Our information theoretic lower bounds rule out this method as a way to approximate the largest root, at least in full generality, since knowing all of the coefficients of a real-rooted polynomial up to a $(1 + 1/\operatorname{poly}(n))$ error for any $\operatorname{poly}(n)$ is no better than just knowing the first $\log n$ coefficients exactly, in the worst case, even under the promise that the given polynomials have a common interlacing. In other words, even an MCMC oracle that gives $1 + 1/\operatorname{poly}(n)$ approximation of all coefficients would not generically allow one to round an interlacing family of depth greater than logarithmic in $n$, since the error accumulated at each step would be $1/\operatorname{polylog}(n)$.

**Connections to Poisson Binomial Distributions** Finally, there is a probabilistic view of Problem 1.1. Assume that $X = B(p_1) + \cdots + B(p_n)$ is a sum of independent Bernoulli random variables, i.e. a Poisson binomial, with parameters $p_1, \ldots, p_n \in [0, 1]$. Then Problem 1.1 becomes the following: Given the first $k$ moments of $X$ how well can we approximate $\max_i p_i$? In this view, our paper is related to [DP15], where it was shown that any pair of such Poisson binomial random variables with the same first $k$ moments have total variation distance at most $2^{-\Omega(k)}$. However, the bound on the total variation distance does not directly imply a bound on the maximum $p_i$.

**Discussion** Besides conducting a precise study of the dependence of the largest root of a real-rooted polynomial on its coefficients, the results of this paper shed light on what a truly efficient algorithm for interlacing families might look like. On one hand, our running time of $2^{\tilde{O}(m^{1/3})}$ shows that the problem is not ETH hard, and is unnatural enough to suggest that a faster algorithm (for instance, quasipolynomial) may exist. On the other hand, our lower bounds show that the polynomials that arise in this method are in general hard to compute in a rather robust sense: namely, obtaining an inverse polynomial error approximation of their largest roots requires knowing *many* coefficients *exactly*. This implies that in order to obtain an efficient algorithm for even approximately simulating the interlacing families proof technique, one will have to exploit finer properties of the polynomials at hand, or find a more "global" proof which is able to reason about the error in a more sophisticated amortized manner, or perhaps track a more well-conditioned quantity in place of the largest root, which can be computed using fewer coefficients and which still satisfies an approximate interlacing property.

# 2 Preliminaries

We let $[n]$ denote the set $\{1, \ldots, n\}$. We use the notation $\binom{[n]}{k}$ to denote the family of subsets $T \subseteq [n]$ with $|T| = k$. We let $S_n$ denote the set of permutations on $[n]$, i.e. the set of bijections $\sigma : [n] \to [n]$.

We use bold letters to denote vectors. For a vector $\boldsymbol{\mu} \in \mathbb{R}^n$, we denote its coordinates by $\mu_1, \ldots, \mu_n$. We let $\mu_{\max}$ and $\mu_{\min}$ denote $\max_i \mu_i$ and $\min_i \mu_i$ respectively.

For a symmetric matrix $A$, we denote the vector of eigenvalues of $A$, i.e. the roots of $\det(xI - A)$, by $\boldsymbol{\lambda}(A)$. Similarly we denote the largest and smallest eigenvalues by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$. We slightly abuse notation, and for a polynomial $p$ we write $\boldsymbol{\lambda}(p)$ to denote the vector of roots of $p$. We also write $\lambda_{\max}(p)$ to denote the largest root of $p$.

For a graph $G = (V, E)$ we let $\deg_{\max}(G)$ denote the maximum degree of its vertices and $\deg_{\text{avg}}(G)$ denote the average degree of its vertices, i.e. $2|E|/|V|$.

What follows are mostly standard facts; the proofs of Fact 2.4, Fact 2.13, Fact 2.13, and Fact 2.22 are included in Appendix A for completness.

**Facts from Linear Algebra** For a matrix $A \in \mathbb{R}^{n \times n}$, the characteristic polynomial of $A$ is defined as $\det(xI - A)$. Letting $\sigma_k(A)$ be the sum of all principal $k$-by-$k$ minors of $A$, we have:

$$\det(xI - A) = \sum_{k=0}^{n} x^{n-k} \sigma_k(A).$$

There are several algorithms that for a matrix $A \in \mathbb{R}^{n \times n}$ calculate $\det(xI - A)$ in time polynomial in $n$. By the above identity, we can use any such algorithm to efficiently obtain $\sigma_k(A)$ for any $1 \le k \le n$.

The following proposition is proved in [MSS13b] using the Cauchy-Binet formula.

**Proposition 2.1.** *Let $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$. Then,*

$$\det\left(xI - \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T\right) = \sum_{k=0}^{n} (-1)^k x^{n-k} \sum_{S \subseteq \binom{[m]}{k}} \sigma_k\left(\sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^T\right).$$

**Symmetric Polynomials** We will make heavy use of the elementary symmetric polynomials, which relate the roots of a polynomial to its coefficients.

**Definition 2.2.** *Let $e_k \in \mathbb{R}[\mu_1, \ldots, \mu_n]$ denote the $k$-th elementary symmetric polynomial defined as*

$$e_k(\boldsymbol{\mu}) := \sum_{T \in \binom{[n]}{k}} \prod_{i \in T} \mu_i.$$

**Fact 2.3.** *Consider the monic univariate polynomial* $\chi(x) = x^n + c_1 x^{n-1} + \cdots + c_n$. *Suppose that* $\mu_1, \dots, \mu_n$ *are the roots of* $\chi$. *Then for every* $k \in [n]$,

$$c_k = (-1)^k e_k(\mu_1, \dots, \mu_n).$$

This means that knowing the top $k$ coefficients of a polynomial is equivalent to knowing the first $k$ elementary symmetric polynomials of the roots. It also implies the following fact about how shifting and scaling affect the elementary symmetric polynomials.

**Fact 2.4.** *Let* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n$ *be such that* $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ *for* $i = 1, \dots, k$. *If* $a, b \in \mathbb{R}$ *then* $e_i(a\boldsymbol{\mu} + b) = e_i(a\boldsymbol{\nu} + b)$ *for* $i = 1, \dots, k$.

We will use the following relationship between the elementary symmetric polynomials and the power sum polynomials.

**Theorem 2.5** (Newton's Identities)**.** *For* $1 \le k \le n$, *the polynomial* $p_k(\boldsymbol{\mu}) := \sum_{i=1}^n \mu_i^k$ *can be written as* $q_k(e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu}))$, *where* $q_k \in \mathbb{R}[e_1, \dots, e_k]$. *Furthermore,* $q_k$ *can be computed at any point in time* $\mathrm{poly}(k)$.

One of the immediate corollaries of the above is the following.

**Corollary 2.6.** *Let* $p(x) \in \mathbb{R}[x]$ *be a univariate polynomial with* $\deg p \le k$. *Then* $\sum_{i=1}^n p(\mu_i)$ *can be written as*

$$q(e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})),$$

*where* $q \in \mathbb{R}[e_1, \dots, e_k]$. *Furthermore,* $q$ *can be computed at any point in time* $\mathrm{poly}(k)$.

Theorem 2.5 shows how $p_1, \dots, p_k$ can be computed from $e_1, \dots, e_k$. The reverse is also true. A second set of identities, also known as Newton's identities, imply the following.

**Theorem 2.7** (Newton's Identities)**.** *For each* $k \in [n]$, $e_k(\boldsymbol{\mu})$ *can be written as a polynomial of* $p_1(\boldsymbol{\mu}), \dots, p_k(\boldsymbol{\mu})$ *which can be computed in time* $\mathrm{poly}(k)$.

A corollary of the above and Theorem 2.5 is the following.

**Corollary 2.8.** *For two vectors* $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n$, *we have*

$$(\forall i \in [k] : e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})) \iff (\forall i \in [k] : p_i(\boldsymbol{\mu}) = p_i(\boldsymbol{\nu}))\,.$$

**Chebyshev Polynomials** Chebyshev polynomials of the first kind, which we will simply call Chebyshev polynomials, are defined as follows.

**Definition 2.9.** *Let the polynomials* $T_0, T_1, \dots \in \mathbb{R}[x]$ *be recursively defined as*

$$\begin{aligned} T_0(x) &:= 1, \\ T_1(x) &:= x, \\ T_{n+1}(x) &:= 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

*We will call* $T_k$ *the* $k$*-th Chebyshev polynomial.*

Notice that the coefficients of $T_k$ can be computed in $\text{poly}(k)$ time, by the above recurrence for example. Chebyshev polynomials have many useful properties, some of which we mention below. For further information, see [Sze39].

**Fact 2.10.** *For $k \geq 0$ and $\theta \in \mathbb{R}$, we have*

$$T_k(\cos(\theta)) = \cos(k\theta),$$
$$T_k(\cosh(\theta)) = \cosh(k\theta).$$

**Fact 2.11.** *The $k$-th Chebyshev polynomial $T_k$ has degree $k$.*

**Fact 2.12.** *For any $x \in [-1, 1]$, we have $T_k(x) \in [-1, 1]$.*

**Fact 2.13.** *For any integer $k \geq 0$, $T_k(1 + x)$ is monotonically increasing for $x \geq 0$. Furthermore for $x \geq 0$,*

$$T_k(1 + x) \geq (1 + \sqrt{2x})^k/2.$$

In our approximate lower bound we will use the following connection between Chebyshev polynomials and graphs, due to Godsil and Gutman [**?**].

**Fact 2.14.** *If $A_n$ is the adjacency matrix of a cycle on $n$ vertices, then*

$$\det(2xI - A_n) = 2T_n(x).$$

**Graphs with Large Girth** In order to prove some of our impossibility results, we use the existence of extremal graphs with no small cycles.

**Definition 2.15.** *For an undirected graph $G$, we denote the length of its shortest cycle by $\text{girth}(G)$. If $G$ is a forest, then $\text{girth}(G) = \infty$.*

The following conjecture by Erdős characterizes extremal graphs with no small cycles.

**Conjecture 2.16** (Erdős's girth conjecture [Erd64])**.** *For every integer $k \geq 1$ and sufficiently large $n$, there exist graphs $G$ on $n$ vertices with $\text{girth}(G) > 2k$ that have $\Omega(n^{1+1/k})$ edges, or in other words satisfy $\deg_{\text{avg}}(G) = \Omega(n^{1/k})$.*

This conjecture has been proven for $k = 1, 2, 3, 5$ [Wen91]. We will use the following more general construction of graphs of somewhat lower girth.

**Theorem 2.17** ([LU95])**.** *If $d$ is a prime power and $t \geq 3$ is odd, there is a $d$-regular bipartite graph $G$ on $2d^t$ vertices with $\text{girth}(G) \geq t + 5$.*

**Signed Adjacency Matrices** Our lower bounds will also utilize facts about signings of graphs.

**Definition 2.18.** *For a graph $G = ([n], E)$, we define a signing to be any function $s : E \to \{-1, +1\}$. We define the signed adjacency matrix $A_s \in \mathbb{R}^{n \times n}$, associated with signing $s$, as follows*

$$A_s(u, v) := \begin{cases} 0 & \{u, v\} \notin E, \\ s(\{u, v\}) & \{u, v\} \in E. \end{cases}$$

Note that by definition, $A_s$ is symmetric and has zeros on the diagonal. The following fact is immediate.

**Fact 2.19.** *For a signed adjacency matrix $A_s$ of a graph $G$, the eigenvalues $\boldsymbol{\lambda}(A_s)$, i.e. the roots of $\chi(x) := \det(xI - A_s)$, are real. If $G$ is bipartite, the eigenvalues are symmetric about the origin (counting multiplicities).*

Signed adjacency matrices were used in [MSS13a] to prove the existence of bipartite Ramanujan graphs of all degrees. We state one of the main results of [MSS13a] below.

**Theorem 2.20** ([MSS13a])**.** *For every graph $G$, there exists a signing $s$ such that*

$$\lambda_{\max}(A_s) \leq 2\sqrt{\deg_{\max}(G) - 1}.$$

By Fact 2.19, we have the following immediate corollary.

**Corollary 2.21.** *For every bipartite graph $G$, there exists a signing $s$ such that the eigenvalues of $A_s$ have absolute value at most $2\sqrt{\deg_{\max}(G) - 1}$.*

We note that trivially signing every edge with $+1$ is often far from achieving the above bound as witnessed by the following fact.

**Fact 2.22.** *Let $A$ be the adjacency matrix of a graph $G = ([n], E)$ (i.e. the signed adjacency matrix where the sign of every edge is $+1$). Then the maximum eigenvalue of $A$ is at least $\deg_{\text{avg}}(G)$.*

## 3 Approximation of the Largest Root

In this section we give an answer to Problem 1.1. As witnessed by Fact 2.3, knowing the top $k$ coefficients of the polynomial $\chi_{\boldsymbol{\mu}}$ is the same as knowing $e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})$. Therefore, without loss of generality and more conveniently, we state the results in terms of knowing $e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})$.

**Theorem 3.1.** *There is an algorithm that receives $n$ and $e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})$ for some unknown $\boldsymbol{\mu} \in \mathbb{R}_+^n$ as input and outputs $\mu^*_{\max}$, an approximation of $\mu_{\max}$, with the guarantee that*

$$\mu^*_{\max} \leq \mu_{\max} \leq \alpha_{k,n} \cdot \mu^*_{\max},$$

*where the approximation factor $\alpha_{k,n}$ is*

$$\alpha_{k,n} = \begin{cases} n^{1/k} & k \leq \log n, \\ 1 + O(\frac{\log n}{k})^2 & k > \log n. \end{cases}$$

*Furthermore the algorithm runs in time $\text{poly}(k)$.*

Note that there is a change in the behavior of the approximation factor in the two regimes $k \ll \log n$ and $k \gg \log n$. When $k > \log n$, the expression $n^{1/k}$ is $1 + \Theta(\frac{\log n}{k})$ which can be a much worse bound compared to $1 + O(\frac{\log n}{k})^2$. When $k$ is near the threshold of $\log n$, $n^{1/k}$ and $1 + \Theta(\frac{\log n}{k})^2$ are close to each other and both of the order of $1 + \Theta(1)$.

We complement this result by showing information-theoretic lower bounds.

**Theorem 3.2.** *For every $1 \le k < n$, there are two vectors $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n_+$ such that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \dots, k$, and*

$$\frac{\nu_{\max}}{\mu_{\max}} \ge \beta_{k,n},$$

*where*

$$\beta_{k,n} = \begin{cases} n^{\Omega(1/k)} & k \le \log n, \\ 1 + \Omega\left(\frac{\log \frac{2n}{k}}{k}\right)^2 & k > \log n. \end{cases}$$

This shows that no algorithm can approximate $\mu_{\max}$ by a factor better than $\beta_{k,n}$ using $e_1(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})$. Note that for $k < n^{1-\epsilon}$, $\beta_{k,n} = \alpha^c_{k,n}$ for some constant $c$ bounded away from zero. For constant $k$, it is possible to give a constant multiplicative bound assuming Erdős's girth conjecture.

**Theorem 3.3.** *Assume that $k$ is fixed and Erdős's girth conjecture (Conjecture 2.16) is true for graphs of girth $> 2k$. Then for large enough $n$ there are two vectors $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n_+$ such that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \dots, k$ and*

$$\frac{\nu_{\max}}{\mu_{\max}} \ge \Omega(n^{1/k}).$$

## 3.1 Proof of Theorem 3.1: An Algorithm for Approximating the Largest Root

We consider two cases: if $k \le \log n$ we return $(p_k(\boldsymbol{\mu})/n)^{1/k}$ as the estimate for the maximum root. It is not hard to see that in this case, $(p_k(\boldsymbol{\mu})/n)^{1/k}$ gives an $n^{1/k}$ approximation of the maximum root (see Claim 3.4 below).

For $k > \log n$ we can still use $(p_k(\boldsymbol{\mu})/n)^{1/k}$ to estimate the maximum root, but this only guarantees a $1 + O(\frac{\log n}{k})$ approximation. We show that using the machinery of Chebyshev polynomials we can obtain a better bound. The pseudocode for the algorithm can be see in Algorithm 1.

**Algorithm 1** Algorithm For Approximating the Maximum Root From Top Coefficients

**Input:** $n$ and $e_1(\boldsymbol{\mu}), e_2(\boldsymbol{\mu}), \dots, e_k(\boldsymbol{\mu})$ for some $\boldsymbol{\mu} \in \mathbb{R}^n_+$.
**Output:** $\mu^*_{\max}$, an approximation of $\mu_{\max}$.

```
if k ≤ log n then
    Compute p_k(μ) = Σ_{i=1}^n μ_i^k using Newton's identities (Theorem 2.5).
    return (p_k(μ)/n)^{1/k}.
else
    t ← e_1(μ).
    loop
        Compute p(μ) := Σ_{i=1}^n T_k(μ_i/t) using Corollary 2.6.
        if p(μ) > n then
            return μ*_max ← t.
        end if
        t ← t / (1 + (20 log n / k)^2).
    end loop
end if
```

We will prove the following claims to show the correctness of Algorithm 1. Let us start with the case $k \le \log n$.

**Claim 3.4.** *For any $k \geq 1$ we have*

$$\left(\frac{p_k(\boldsymbol{\mu})}{n}\right)^{1/k} \leq \mu_{\max} \leq p_k(\boldsymbol{\mu})^{1/k}.$$

*Proof.* Observe,

$$p_k(\boldsymbol{\mu})/n = \sum_{i=1}^{n} \frac{\mu_i^k}{n} \leq \sum_{i=1}^{n} \frac{\mu_{\max}^k}{n} = \mu_{\max}^k \leq \sum_{i=1}^{n} \mu_i^k = p_k(\boldsymbol{\mu}),$$

Taking $\frac{1}{k}$-th root of all sides of the above proves the claim. □

The rest of the section handles the case where $k > \log n$. Our first claim shows that as long as $t \geq \mu_{\max}$, the algorithm keeps decreasing $t$ by a multiplicative factor of $(1 - \Omega(\log(n)/k)^2)$. Since at the beginning we have $t = e_1(\boldsymbol{\mu}) \geq \mu_{\max}$, we will have $\mu^*_{\max} \leq \mu_{\max}$.

**Claim 3.5.** *For any $t \geq \mu_{\max}$,*

$$\sum_{i=1}^{n} T_k(\frac{\mu_i}{t}) \leq n.$$

*Proof.* If $t \geq \mu_{\max}$, then $\mu_i/t \in [0,1]$ for every $i \in [n]$. By Fact 2.12, we have

$$\sum_{i=1}^{n} T_k(\frac{\mu_i}{t}) \leq \sum_{i=1}^{n} 1 = n.$$

□

To finish the proof of correctness it is enough to show that $\mu_{\max} \leq \mu^*_{\max}(1 + O(\log n/k)^2)$. This is done in the next claim. It shows that as soon as $t$ gets lower than $\mu_{\max}$, within one more iteration of the loop, the algorithm terminates.

**Claim 3.6.** *For $k > \log n$ and $t > 0$, if $\mu_{\max} > (1 + (\frac{20 \log n}{k})^2)t$, then*

$$\sum_{i=1}^{n} T_k(\frac{\mu_i}{t}) > n.$$

*Proof.* When $\mu_{\max} > (1 + (\frac{20 \log n}{k})^2)t$, by Fact 2.13 we have

$$\begin{aligned} T_k(\frac{\mu_{\max}}{t}) &\geq T_k\left(1 + \left(\frac{20 \log n}{k}\right)^2\right) \geq \frac{1}{2}\left(1 + \sqrt{2} \cdot \frac{20 \log n}{k}\right)^k \\ &\geq \frac{1}{2} \exp\left(\frac{3 \log n}{k}\right)^k > 2n, \end{aligned}$$

where we used the inequality $1 + \sqrt{800}x \geq e^{3x}$ for $x \in [0,1]$.

Now we have

$$\sum_{i=1}^{n} T_k(\frac{\mu_i}{t}) = T_k(\frac{\mu_{\max}}{t}) + \sum_{i \neq \operatorname{argmax}_j \mu_j} T_k(\frac{\mu_i}{t}) \geq 2n - (n-1) > n,$$

where we used Fact 2.12 and Fact 2.13 to conclude $T_k(\frac{\mu_i}{t}) \geq -1$ for every $i$. □

The above claim also gives us a bound on the number of iterations in which the algorithm terminates. This is because we start the loop with $t = e_1(\boldsymbol{\mu}) \leq n\mu_{\max}$ and the loop terminates within one iteration as soon as $t < \mu_{\max}$. Therefore the number of iterations is at most

$$1 + \frac{\log n}{\log\left(1 + (\frac{20\log n}{k})^2\right)} = O\left(\log n \cdot (\frac{k}{\log n})^2\right) = O(k^2).$$

### 3.2 Proofs of Theorems 3.2 and 3.3: Matching Lower Bounds

The machinery of Chebyshev polynomials was used to prove Theorem 3.1. We show that this machinery can also be used to prove a weaker version of Theorem 3.2.

**Theorem 3.7.** *For every $1 \leq k < n$, there are $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^n$ such that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \dots, k$ and*

$$\frac{\nu_{\max}}{\mu_{\max}} \geq 1 + \Omega(1/k^2)$$

*Proof.* First let us prove this for $k = n - 1$. Let $\boldsymbol{\mu}$ be the set of roots of $T_n(x-1)+1$ and $\boldsymbol{\nu}$ the set of roots of $T_n(x-1)-1$. These two polynomials are the same except for the constant term. It follows that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \dots, n-1$. We use the following lemma to prove that $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^n$.

**Lemma 3.8.** *For $\theta \in \mathbb{R}$, the roots of $T_n(x) - \cos(\theta)$, counting multiplicities, are $\cos(\frac{\theta+2\pi i}{n})$ for $i = 0, \dots, n-1$.*

*Proof.* We have

$$T_n\left(\cos(\frac{\theta + 2\pi i}{n})\right) = \cos(\theta + 2\pi i) = \cos(\theta).$$

For almost all $\theta$ these roots are distinct and since $T_n$ has degree $n$, it follows that they are all of the roots. When some of these roots collide, we can perturb $\theta$ and use the fact that roots are continuous functions of the polynomial coefficients to prove the statement. □

Using the above lemma for $\theta = \pi$ and $\theta = 0$, we get that $\mu_i = 1+\cos(\frac{\pi+2\pi i}{n})$ and $\nu_i = 1+\cos(\frac{2\pi i}{n})$. This proves that $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}_+^n$. Moreover we have

$$\frac{\nu_{\max}}{\mu_{\max}} = \frac{1+1}{1+\cos(\pi/n)} = 1 + \Omega(1/n^2).$$

This finishes the proof for $k = n - 1$.

Now let us prove the statement for general $k$. By applying the above proof for $n = k + 1$, we get $\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\nu}} \in \mathbb{R}_+^{k+1}$ such that $e_i(\tilde{\boldsymbol{\mu}}) = e_i(\tilde{\boldsymbol{\nu}})$ for $i = 1, \dots, k$ and

$$\frac{\tilde{\nu}_{\max}}{\tilde{\mu}_{\max}} \geq 1 + \Omega(1/k^2).$$

Now construct $\boldsymbol{\mu}, \boldsymbol{\nu}$ from $\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\nu}}$ by adding zeros to make the total count $n$. It is not hard to see, by using Corollary 2.8 that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \dots, k$. Moreover $\boldsymbol{\mu}_{\max} = \tilde{\boldsymbol{\mu}}_{\max}$ and $\boldsymbol{\nu}_{\max} = \tilde{\boldsymbol{\nu}}_{\max}$. This finishes the proof. □

Note that the above lower bound is the same as the lower bound in Theorem 3.2 when $k = \Omega(n)$. However, to prove Theorem 3.2 and Theorem 3.3 we need more tools. The crucial idea we use to get the stronger Theorem 3.2 and Theorem 3.3 is the following observation about signed adjacency matrices for graphs of large girth.

**Lemma 3.9.** *Let $G = ([n], E)$ be a graph and $D \in \mathbb{R}^{n\times n}$ an arbitrary diagonal matrix. If $\operatorname{girth}(G) > k$, then the top $k$ coefficients of the polynomial $\chi(x) = \det(xI - (D + A_s))$ are independent of the signing $s$. In other words,*

$$e_i(\boldsymbol{\lambda}(D + A_{s_1})) = e_i(\boldsymbol{\lambda}(D + A_{s_2})),$$

*for any two signings $s_1, s_2$ and $i = 1, \ldots, k$.*

We will apply the above lemma, with $D = 0$, to graphs of large girth constructed based on Conjecture 2.16 or Theorem 2.17, in order to prove Theorem 3.3 and Theorem 3.2 for the regime $k \leq \Theta(\log n)$. In order to prove Theorem 3.3 for the regime $k \geq \Theta(\log n)$, we marry these constructions with Chebyshev polynomials. We will prove the above lemma at the end of this section, after proving Theorem 3.2 and Theorem 3.3.

First let us prove Theorem 3.3.

*Proof of Theorem 3.3.* We apply Lemma 3.9 to the following graph construction, the proof of which we defer to Appendix B.

**Claim 3.10.** *Let $k$ be fixed and assume that Conjecture 2.16 is true for graphs of girth $> 2k$. Then, for all sufficiently large $n$, there exist bipartite graphs $G = ([n], E)$ with $\operatorname{girth}(G) > 2k$, $\deg_{\max}(G) = O(n^{1/k})$, and $\deg_{\text{avg}}(G) = \Omega(n^{1/k})$.*

Let $G$ be the graph from the above claim. Let $s_1$ be the trivial signing that assigns $+1$ to every edge, and $s_2$ the signing guaranteed by Corollary 2.21. Now let $\boldsymbol{\nu} = \boldsymbol{\lambda}(A_{s_1}^2)$, i.e. the square of the eigenvalues of $A_{s_1}$, and $\boldsymbol{\mu} = \boldsymbol{\lambda}(A_{s_2}^2)$, i.e. the square of the eigenvalues of $A_{s_2}$. By Lemma 3.9 and Corollary 2.8, we have

$$p_i(\boldsymbol{\nu}) = p_{2i}(\boldsymbol{\lambda}(A_{s_1})) = p_{2i}(\boldsymbol{\lambda}(A_{s_2})) = p_i(\boldsymbol{\mu}),$$

for $i = 1, \ldots, k$. By another application of Corollary 2.8, we have $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \ldots, k$.

On the other hand, by Corollary 2.21, we have

$$\mu_{\max} = \lambda_{\max}(A_{s_2})^2 \leq 4(\deg_{\max}(G) - 1) = O(n^{1/k}),$$

and by Fact 2.22, we have

$$\nu_{\max} = \lambda_{\max}(A_{s_1})^2 \geq \deg_{\text{avg}}(G)^2 = \Omega(n^{2/k}).$$

Therefore

$$\frac{\nu_{\max}}{\mu_{\max}} = \frac{\Omega(n^{2/k})}{O(n^{1/k})} = \Omega(n^{1/k}).$$

□

Now let us prove Theorem 3.2 for $k \leq \Theta(\log n)$.

*Proof of Theorem 3.2 for $k \leq \Theta(\log n)$.* We apply Lemma 3.9 to the following graph construction, the proof of which we defer to Appendix B.

**Claim 3.11.** *Let $d$ be a prime. For all sufficiently large $n$, there exist bipartite graphs $G = ([n], E)$ with $\mathrm{girth}(G) = \Omega(\log n)$, $\deg_{\max}(G) \leq d$, and $\deg_{\mathrm{avg}}(G) \geq d/2$.*

We will fix $d$ to a specific prime later. Similar to the proof of Theorem 3.3, we let $s_1$ be the trivial signing with all +1s and $s_2$ be the signing guaranteed by Corollary 2.21. Let $t = \Omega(\log n/k)$ be an even integer such that $tk < \mathrm{girth}(G)$. Such a $t$ exists when $k < c \log n$ for some constant $c$. Take $\boldsymbol{\nu} = \boldsymbol{\lambda}(A_{s_1}^t)$ and $\boldsymbol{\mu} = \boldsymbol{\lambda}(A_{s_2}^t)$. Then we have

$$\mu_{\max} = \lambda_{\max}(A_{s_2})^t \leq (2\sqrt{d-1})^t,$$

and

$$\nu_{\max} = \lambda_{\max}(A_{s_1})^t \geq (d/2)^t.$$

This means that

$$\frac{\nu_{\max}}{\mu_{\max}} \geq \left(\frac{d}{4\sqrt{d-1}}\right)^t \geq e^{\Omega(\log n/k)} = n^{\Omega(1/k)},$$

as long as $\frac{d}{4\sqrt{d-1}} > e$, which happens for sufficiently large $d$ (such as $d = 127$). It only remains to show that $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \ldots, k$. For every $i \in [k]$ we have $t \cdot i < \mathrm{girth}(G)$, which by Lemma 3.9 gives us

$$p_i(\boldsymbol{\nu}) = p_{t \cdot i}(\boldsymbol{\lambda}(A_{s_1})) = p_{t \cdot i}(\boldsymbol{\lambda}(A_{s_2})) = p_i(\boldsymbol{\nu}),$$

and this finishes the proof because of Corollary 2.8. □

The above method unfortunately does not seem to directly extend to the regime $k \geq \Theta(\log n)$, since for large $k$, getting $\mathrm{girth}(G) > k$ requires many vertices of degree at most 2.[1] Instead we use the machinery of Chebyshev polynomials to boost our graph constructions.

*Proof of Theorem 3.2 for $k \geq \Theta(\log n)$.* Since Theorem 3.7 proves the same desired bound as Theorem 3.2 when $k = \Omega(n)$, we may without loss of generality assume that $n/k$ is at least a large enough constant. Let $m$ be the largest integer such that

$$c \cdot \frac{m}{\log m} \leq \frac{n}{k}, \tag{1}$$

where $c$ is a large constant that we will fix later. It is easy to see that

$$m = \Theta\left(\frac{n}{k}\log(\frac{n}{k})\right). \tag{2}$$

We have already proved Theorem 3.2 for the small $k$ regime. Using this proof (for $n = m$ and $k = \Theta(\log m)$), we can find $\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\nu}} \in \mathbb{R}_+^m$ such that $e_i(\tilde{\boldsymbol{\mu}}) = e_i(\tilde{\boldsymbol{\nu}})$ for $i = 1, \ldots, \Omega(\log m)$ and $\tilde{\nu}_{\max}/\tilde{\mu}_{\max} \geq m^{1/\Theta(\log m)} \geq 2$. Without loss of generality, by a simple scaling, we may assume that $\tilde{\nu}_{\max} = 1$ and $\tilde{\mu}_{\max} \leq 1/2$.

Let $\chi_{\tilde{\boldsymbol{\mu}}}(x), \chi_{\tilde{\boldsymbol{\nu}}}(x) \in \mathbb{R}[x]$ be the unique monic polynomials whose roots are $\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\nu}}$ respectively. By construction, the top $\Omega(\log m)$ coefficients of these polynomials are the same. We boost the number of equal coefficients by composing them with Chebyshev polynomials. Let $p(x) := \chi_{\tilde{\boldsymbol{\mu}}}(T_t(x))$ and

[1] Unless all degrees are 2 in which case we can actually reprove Theorem 3.7; we omit the details here.

$q(x) := \chi_{\tilde{\boldsymbol{\nu}}}(T_t(x))$ where $t = \lfloor n/m \rfloor$. Note that $\deg p = \deg q = tm \le n$. We let $\boldsymbol{\mu}, \boldsymbol{\nu}$ be the roots of $p(x), q(x)$ together with some additional zeros to make their counts $n$. First, we show that the top $k$ coefficients of $p, q$ are the same. Then we show that they are real rooted, i.e., $\boldsymbol{\mu}, \boldsymbol{\nu}$ are real vectors. Finally, we lower bound $\nu_{\max}/\mu_{\max}$.

Note that $p(x), q(x)$ have degree $tm$. They are not monic, but their leading terms are the same. Besides the leading terms, we claim that they have the same top $\Omega(t \log m)$ coefficients. This follows from the fact that $T_t(x)$ is a degree $t$ polynomial. When expanding either $\chi_{\tilde{\boldsymbol{\nu}}}(T_t(x))$ or $\chi_{\tilde{\boldsymbol{\mu}}}(T_t(x))$, terms of degree $\le m - \Omega(\log m)$ in $\chi_{\tilde{\boldsymbol{\mu}}}, \chi_{\tilde{\boldsymbol{\nu}}}$ produce monomials of degree at most $tm - \Omega(t \log m)$, which means that the top $\Omega(t \log m)$ coefficients are the same. This shows that the first $\Omega(t \log m)$ elementary symmetric polynomials of $\boldsymbol{\mu}, \boldsymbol{\nu}$ are the same. It follows that the first $k$ elementary symmetric polynomials to be the same, which follows from

$$\Omega(t \log m) = \Omega(\frac{n \log m}{m}) \ge \Omega(ck) \ge k,$$

where for the first inequality we used (1) and for the second inequality we assumed $c$ is large enough that it cancels the hidden constants in $\Omega$.

It is not obvious if $\boldsymbol{\mu}, \boldsymbol{\nu}$ are even real. This is where we crucially use the properties of the Chebyshev polynomial $T_t(x)$. Note that the roots of $\chi_{\tilde{\boldsymbol{\mu}}}(x)$ and $\chi_{\tilde{\boldsymbol{\nu}}}(x)$, i.e. $\tilde{\mu}_i$'s and $\tilde{\nu}_i$'s are all in $[0, 1] \subseteq [-1, 1]$. Therefore each one of them can be written as $\cos(\theta)$ for some $\theta \in \mathbb{R}$. By Lemma 3.8 the equation

$$T_t(x) = \cos(\theta),$$

has $t$ real roots (counting multiplicities), and they are simply $x = \cos(\frac{\theta + 2\pi i}{t})$ for $i = 0, \ldots, t-1$. So for each root of $\chi_{\tilde{\boldsymbol{\mu}}}(x)$ we have $t$ roots of $\chi_{\tilde{\boldsymbol{\mu}}}(T_t(x))$, all in $[-1, 1]$. This means that all of the roots of $p(x)$ are real and in $[-1, 1]$. By a similar argument, all of the roots of $q(x)$ are real and in $[-1, 1]$.

The largest root of $q(x)$ is 1, since

$$q(1) = \chi_{\tilde{\boldsymbol{\nu}}}(T_t(1)) = \chi_{\tilde{\boldsymbol{\nu}}}(1) = \chi_{\tilde{\boldsymbol{\nu}}}(\nu_{\max}) = 0.$$

On the other hand, the largest root of $p(x)$ is at most $\cos(\pi/3t)$, because for any $x \in (\cos(\pi/3t), 1]$ there is $\theta \in [0, \pi/3t)$ such that $x = \cos(\theta)$ and this means

$$p(x) = \chi_{\tilde{\boldsymbol{\mu}}}(T_t(\cos(\theta))) = \chi_{\tilde{\boldsymbol{\mu}}}(\cos(t\theta)) \neq 0,$$

because $\cos(t\theta) > \cos(\pi/3) = 1/2 \ge \tilde{\mu}_{\max}$.

By the above arguments, $\boldsymbol{\mu}, \boldsymbol{\nu}$ satisfy almost all of the desired properties, except that they could be negative. However we know that $\boldsymbol{\mu}, \boldsymbol{\nu} \in [-1, 1]^n$. So using Fact 2.4 we can easily make them nonnegative. We simply replace $\boldsymbol{\mu}, \boldsymbol{\nu}$ by $\boldsymbol{\mu} + 1, \boldsymbol{\nu} + 1$. Then $\boldsymbol{\mu}, \boldsymbol{\nu} \in [0, 2]^n$ and $e_i(\boldsymbol{\mu}) = e_i(\boldsymbol{\nu})$ for $i = 1, \ldots, k$. Finally, we have

$$\frac{\nu_{\max}}{\mu_{\max}} \ge \frac{1+1}{1 + \cos(\pi/3t)} = 1 + \Omega\left(\frac{1}{t^2}\right) = 1 + \Omega\left(\frac{m}{n}\right)^2 = 1 + \Omega\left(\frac{\log \frac{2n}{k}}{k}\right)^2,$$

where we used (2) for the last equality. □

Having finished the proofs of Theorem 3.2, Theorem 3.3 we are ready to prove Lemma 3.9.

*Proof of Lemma 3.9.* By Corollary 2.8, it is enough to prove that

$$p_k(\boldsymbol{\lambda}(D + A_{s_1})) = p_k(\boldsymbol{\lambda}(D + A_{s_2})),$$

for $k < \text{girth}(G)$. This is the same as proving

$$\text{Tr}\left((D + A_{s_1})^k\right) = \text{Tr}\left((D + A_{s_2})^k\right). \tag{3}$$

For a matrix $M \in \mathbb{R}^{n\times n}$ we have the following identity

$$\text{Tr}(M^k) = \sum_{(v_1,\ldots,v_k)\in[n]^k} M_{v_1,v_2} M_{v_2,v_3} \ldots M_{v_{k-1},v_k} M_{v_k,v_1}.$$

For ease of notation let us identify $v_{k+1}$ with $v_1$. We apply the above formula to both sides of (3). The sequence $(v_1, \ldots, v_k)$ can be interpreted as a sequence of vertices in the graph $G$. If for any $i$, $v_i \neq v_{i+1}$ and $\{v_i, v_{i+1}\} \notin E$, then the term inside the sum vanishes. Therefore we can restrict the sum to those terms $(v_1, \ldots, v_k)$ where for each $i \in [k]$, either $v_i = v_{i+1}$ or $v_i$ and $v_{i+1}$ are connected in $G$. To borrow and abuse some notation from Markov chains, let us call such a sequence a lazy closed walk of length $k$. In order to prove (3) it is enough to prove that for any such lazy closed walk we get the same term for both $s_1$ and $s_2$.

Let $(v_1, \ldots, v_k)$ be one such lazy closed walk. Consider a particle that at time $i$ resides at $v_i$. In each step, the particle either does not move or moves to a neighboring vertex, and at time $k+1$ it returns to its starting position. For each step that the particle does not move we get one of the entries of $D$, corresponding to the current vertex, as a factor. This is clearly independent of the signing. When the particle moves however, we get the sign of the edge over which it moved as a factor. We will show that the particle must cross each edge an even number of times. Therefore the signs for each edge cancel each other and we get the same result for $s_1, s_2$.

Consider the induced subgraph on $v_1, \ldots, v_k$. Because $k < \text{girth}(G)$, this subgraph has no cycles; therefore it must be a tree. A (lazy) closed walk crosses any cut in a graph an even number of times. Each edge in this tree constitutes a cut. Therefore the lazy closed walk $(v_1, \ldots, v_k)$ must cross each edge in the tree an even number of times. □

## 3.3 Lower Bound Given Approximate Coefficients

The lower bounds proved in the previous section show that knowing a small number of the coefficients of a polynomial *exactly* is insufficient to obtain a good estimate of its largest root.

In this section we generalize the construction of Theorem 3.2 to provide a satisfying lower bound to Problem 1.2.

**Proposition 3.12.** *For every integer $n > 1$ and $1 < k < n$ there are degree $n$ polynomials $r(x), s(x)$ such that*

1. *All of the coefficients of $r$ and $s$ except for the $2k^{th}$ are exactly equal, and the $2k^{th}$ coefficients are within a multiplicative factor of $1 + \frac{4}{2^{2k}}$.*
2. *The largest root of $r$ is at least $1 + \Omega(1/k^2)$ of the largest root of $s$.*
3. *$r$ and $s$ have a common interlacing.*

*4. $r$ and $s$ are characteristic polynomials of graph Laplacians. Further, these Laplacians correspond to $2-$lifts of a common base graph.*

*Proof.* Let

$$r(x) := 2T_k^2(3/2 - x) \qquad \textit{and} \qquad s(x) := T_{2k}(3/2 - x).$$

Since

$$T_{2k} = 2T_k^2 - 1,$$

these polynomials differ only in their constant terms. Moreover, we have

$$r(0) = 2T_k^2(3/2) \geq (2^{k-1})^2 \qquad \textit{and} \qquad s(0) \geq 2^{2k-1}/2$$

by Fact 2.13. Thus, they agree on the first $2k - 1$ coefficients, and differ by a multiplicative factor of at most

$$\frac{2^{2k-2} + 1}{2^{2k-2}} = 1 + 4/2^{2k}$$

in the $2k^{th}$ coefficient, establishing (1).

To see (2), observe that $r(x)$ has largest root $3/2 + \cos(2\pi/k)$ whereas $s(x)$ has largest root $3/2 + \cos(2\pi/2k)$. Since the difference of these numbers is

$$\cos(2\pi/k) - \cos(2\pi/2k) = \frac{4\pi^2}{2}(\frac{1}{k^2} - \frac{1}{4k^2} + o(1/k^4)),$$

we conclude that their ratio is at least $1 + \Omega(1/k^2)$.

To see (3), observe that the roots of $r(x)$ are

$$3/2 - \cos(2\pi j/k) \quad j = 0, \ldots, k-1 \quad \text{with multiplicity 2}$$

and the roots of $s(x)$ are

$$3/2 - \cos(2\pi j/2k) \quad j = 0, \ldots, 2k-1 \quad \text{with multiplicity 1},$$

whence $r$ and $s$ have a common interlacing according to Definition 4.1.

For (4) we first apply Fact 2.14 to interpret both $r$ and $s$ as characteristic polynomials of cycles. Let $C_k$ denote a cycle of length $k$ and let $C_k \cup C_k$ denote a union of two such cycles. Then we have:

$$\det(2xI - A_{C_k \cup C_k}) = \det(2xI - A_{C_k})^2 = 4T_k(x)^2 = 2r(3/2 - x),$$

and

$$\det(2xI - A_{C_{2k}}) = 2T_{2k}(x) = 2s(3/2 - x),$$

whence

$$r(x) = \frac{1}{2}\det(3I - A_{C_k \cup C_k} - 2xI) = \frac{(-2)^k}{2}\det(xI - ((3/2)I - (1/2)A_{C_k \cup C_k})$$

and

$$s(x) = \frac{1}{2}\det(3I - A_{C_{2k}} - 2xI) = \frac{(-2)^k}{2}\det(xI - ((3/2)I - (1/2)C_{2k}),$$

which are characteristic polynomials of graph Laplacians of weighted graphs with self loops. Note that both graphs are $2-$covers of $C_k$. Since multiplying by constants does not change any of the properties we are interested in, we can ignore them.

Considering $\tilde{r}(x) = x^{n-k}r(x)$ and $\tilde{s}(x) = x^{n-k}s(x)$ yields examples of the desired dimension $n$; note that multiplying by $x^{n-k}$ simply corresponds to adding isolated vertices to the corresponding graphs. □

## 4 Applications to Interlacing Families

In this section we use Theorem 3.1 to give an $2^{\tilde{O}(\sqrt[3]{m})}$ time algorithm for rounding an interlacing family of depth $m$. Let us start by defining an interlacing family.

**Definition 4.1** (Interlacing). *We say that a real rooted polynomial $g(x) = \alpha_0 \prod_{i=1}^{n-1}(x - \alpha_i)$ interlaces a real rooted polynomial $f(x) = \beta_0 \prod_{i=1}^{n}(x - \beta_i)$ if*

$$\beta_1 \leq \alpha_1 \leq \beta_2 \leq \alpha_2 \leq \cdots \leq \alpha_{n-1} \leq \beta_n.$$

We say that polynomials $f_1, \dots, f_k$ have a common interlacing if there is a polynomial $g$ such that $g$ interlaces all $f_i$. The following key lemma is proved in [MSS13a].

**Lemma 4.2.** *Let $f_1, \dots, f_k$ be polynomials of the same degree that are real rooted and have positive leading coefficients. If $f_1, \dots, f_k$ have a common interlacing, then there is an $i$ such that*

$$\lambda_{\max}(f_i) \leq \lambda_{\max}(f_1 + \cdots + f_k).$$

**Definition 4.3** (Interlacing Family). *Let $S_1, \dots, S_m$ be finite sets. Let $\mathcal{F} \subseteq S_1 \times S_2 \times \cdots \times S_m$ be nonempty. For any $s_1, s_2, \dots, s_m \in \mathcal{F}$, let $f_{s_1,\dots,s_m}(x)$ be a real rooted polynomial of degree $n$ with a positive leading coefficient. For $s_1, \dots, s_k \in S_1 \times \cdots \times S_k$ with $k < m$, let*

$$\mathcal{F}_{s_1,\dots,s_k} := \{t_1, \dots, t_m \in \mathcal{F} : s_i = t_i, \forall 1 \leq i \leq k\}.$$

*Note that $\mathcal{F} = \mathcal{F}_\emptyset$. Define*

$$f_{s_1,\dots,s_k} = \sum_{t_1,\dots,t_m \in \mathcal{F}_{s_1,\dots,s_k}} f_{t_1,\dots,t_m},$$

*and*

$$f_\emptyset = \sum_{t_1,\dots,t_m \in \mathcal{F}} f_{t_1,\dots,t_m}.$$

*We say polynomials $\{f_{s_1,\dots,s_m}\}_{s_1,\dots,s_m \in \mathcal{F}}$ form an* interlacing family *if for all $0 \leq k < m$ and all $s_1, \dots, s_k \in S_1 \times \cdots \times S_k$ the following holds: The polynomials $f_{s_1,\dots,s_k,t_i}$ which are not identically zero have a common interlacing.*

In the above definition we say $m$ is the depth of the interlacing families. It follows by repeated applications of Lemma 4.2 that for any interlacing family $\{f_{s_1,\dots,s_m}\}_{s_1,\dots,s_m \in \mathcal{F}}$, there is a polynomial $f_{s_1,\dots,s_m}$ such that the largest root of $f_{s_1,\dots,s_m}$ is at most the largest root of $f_\emptyset$ [MSS13b, Thm 3.4]. For an $\alpha > 1$, an $\alpha$-approximation *rounding* algorithm for an interlacing family $\{f_{s_1,\dots,s_m}\}_{s_1,\dots,s_m \in \mathcal{F}}$ is an algorithm that returns a polynomial $f_{s_1,\dots,s_m}$ such that

$$\lambda_{\max}(f_{s_1,\dots,s_m}) \leq \alpha \lambda_{\max}(f_\emptyset).$$

Next, we design such a rounding algorithm, given an oracle that computes the first $k$ coefficients of the polynomials in an interlacing family.

**Theorem 4.4.** *Let $S_1, \dots, S_m$ be finite sets and let $\{f_{s_1,\dots,s_m}\}_{s_1,\dots,s_m \in \mathcal{F}}$ be an interlacing family of degree $n$ polynomials. Suppose that we are given an oracle that for any $1 \leq k \leq n$ and $s_1, \dots, s_\ell \in S_1, \dots, S_\ell$ with $\ell < m$ returns the top $k$ coefficients of $f_{s_1,\dots,s_m}$ in time $T(k)$. Then, there is an algorithm that for any $\epsilon > 0$ returns a polynomial $f_{s_1,\dots,s_m}$ such that the largest root of $f_{s_1,\dots,s_m}$ is at most $1 + \epsilon$ times the largest root of $f_\emptyset$, in time $T(O(\log(n) m^{1/3}/\sqrt{\epsilon})) \max\{|S_i|\}^{O(m^{1/3})} \operatorname{poly}(n)$.*

*Proof.* Let $M = m^{1/3}$ and $k \asymp \frac{1}{\sqrt{\epsilon}} \log(n) M$. Then, by Theorem 3.1, for any polynomial $f_{s_1,\dots,s_\ell}$ we can find a $1 + \frac{\epsilon}{2M^2}$ approximation of the largest root of $f_{s_1,\dots,s_m}$ in time $T(k)\,\mathrm{poly}(n)$. We round the interlacing family in $M^2$ many steps and in each step we round $M$ of the coordinates. We make sure that each step only incurs a (multiplicative) approximation of $1 + \frac{\epsilon}{2M^2}$ so that the cumulative approximation error is no more than

$$(1 + \frac{\epsilon}{2M^2})^{M^2} \le 1 + \epsilon$$

as desired.

Let us describe the algorithm inductively. Suppose we have selected $s_1, \dots, s_\ell$ for some $0 \le \ell < m$. We brute force over all polynomials $f_{s_1,\dots,s_\ell,t_{\ell+1},\dots,t_{\ell+M}}$ which are not identically zero for all $t_{\ell+1}, \dots, t_{\ell+M} \in S_{\ell+1}, \dots, S_{\ell+M}$. Note that there are at most $(\max_i |S_i|)^M$ many such polynomials. For any polynomial $f_{s_1,\dots,s_\ell,t_{\ell+1},\dots,t_{\ell+M}}$ (which is not identically zero) we compute $\mu^*_{s_1,\dots,s_\ell,t_{\ell+1},\dots,t_{\ell+M}}$, a $1 + \frac{\epsilon}{2M^2}$ approximation of its largest root using its top $k$ coefficients. We let

$$s_{\ell+1}, \dots, s_{\ell+M} = \operatorname*{argmin}_{t_{\ell+1},\dots,t_{\ell+M}} \mu^*_{s_1,\dots,s_\ell,t_{\ell+1},\dots,t_{\ell+M}}.$$

It follows that the algorithm runs in time $T(k)\,\mathrm{poly}(n)(\max_i |S_i|)^{O(M)}$. Because we have an interlacing family, there is a polynomial $f_{s_1,\dots,s_\ell,t_{\ell+1},\dots,t_{\ell+M}}$ whose largest root is at most the largest root of $f_{s_1,\dots,s_\ell}$, in each step of the algorithm. Therefore,

$$\lambda_{\max}(f_{s_1,\dots,s_{\ell+M}}) \le (1 + \frac{\epsilon}{2M^2})\lambda_{\max}(f_{s_1,\dots,s_\ell}).$$

Therefore, by induction,

$$\lambda_{\max}(f_{s_1,\dots,s_m}) \le (1+\epsilon)\lambda_{\max}(f_\emptyset)$$

as desired. □

We remark that without the use of Chebyshev polynomials and Theorem 3.1, one can obtain the somewhat worse running time of $2^{\tilde{O}(\sqrt{m})}$ by applying the same trick of rounding the vectors in groups rather than one at a time.

To use the above theorem, we need to construct the aforementioned oracle for each application of the interlacing families. Next, we construct such an oracle for several examples.

### 4.1 Oracle for Kadison-Singer

We start with interlacing families corresponding to the Weaver's problem which is an equivalent formulation of the Kadison-Singer problem. Marcus, Spielman, and Srivastava proved the following theorem.

**Theorem 4.5** ([MSS13b]). *Given vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ in isotropic position,*

$$\sum_{i=1}^m \mathbf{v}_i \mathbf{v}_i^T = I,$$

*such that* $\max_i \|\mathbf{v}_i\|^2 \le \delta$, *there is a partitioning* $S_1, S_2$ *of* $[m]$ *such that for* $j \in \{1, 2\}$,

$$\left\| \sum_{i \in S_j} \mathbf{v}_i \mathbf{v}_i^T \right\| \le 1/2 + O(\sqrt{\delta}).$$

We give an algorithm that finds the above partitioning and runs in time $2^{m^{1/3}/\delta}$. It follows from the proof of [MSS13b] that it is enough to design a $(1+\delta)$-approximation rounding algorithm for the following interlacing family.

Let $\mathbf{r}_1, \ldots, \mathbf{r}_m \in \mathbb{R}^n$ be independent random vectors where for each $i$, $S_i$ is the support of $\mathbf{r}_i$. For any $\mathbf{r}_1, \ldots, \mathbf{r}_m \in S_1 \times \ldots S_m$ let

$$f_{\mathbf{r}_1,\ldots,\mathbf{r}_m}(x) = \det(xI - \sum_{i=1}^m \mathbf{r}_i \mathbf{r}_i^T).$$

Marcus et al. [MSS13b] showed that $\{f_{\mathbf{r}_1,\ldots,\mathbf{r}_m}\}_{\mathbf{r}_1,\ldots,\mathbf{r}_m \in S_1 \times \cdots \times S_m}$ is an interlacing family. Next we design an algorithm that returns the first $k$ coefficients of any polynomial in this family in time $(m \cdot \max_i |S_i|)^k$.

**Theorem 4.6.** *Given independent random vectors $\mathbf{r}_1, \ldots, \mathbf{r}_m \in \mathbb{R}^d$ with support $S_1, \ldots, S_m$. There is an algorithm that for any $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in S_1 \times \cdots \times S_\ell$ with $1 \le \ell \le m$ and $1 \le k \le n$ returns the top $k$ coefficients of $f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell}$ in time $(m \cdot \max_i |S_i|)^k \operatorname{poly}(n)$.*

*Proof.* Fix $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in S_1 \times \cdots \times S_\ell$ for some $1 \le \ell \le m$. It is sufficient to show that for any $0 \le k \le n$, we can compute the coefficient of $x^{n-k}$ of $f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell}$ in time $(m \cdot \max_i |S_i|)^k \operatorname{poly}(n)$. First, observe that

$$f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell} = \mathbb{E}_{\mathbf{r}_{\ell+1},\ldots,\mathbf{r}_m}\left[\det\left(xI - \sum_{i=1}^m \mathbf{r}_i \mathbf{r}_i^T\right)\right].$$

So, by Proposition 2.1, the coefficient of $x^{n-k}$ in the above is equal to

$$(-1)^k \sum_{T \subseteq \binom{[m]}{k}} \mathbb{E}_{\mathbf{r}_{\ell+1},\ldots,\mathbf{r}_m}\left[\sigma_k\left(\sum_{i\in T} \mathbf{r}_i \mathbf{r}_i^T\right)\right].$$

Note that there are at most $m^k$ terms in the above summation. For any $T \subseteq \binom{[m]}{k}$, we can exactly compute $\mathbb{E}_{\mathbf{r}_{\ell+1},\ldots,\mathbf{r}_m}\left[\sigma_k(\sum_{i\in T} \mathbf{r}_i \mathbf{r}_i^T)\right]$ in time $(\max_i |S_i|)^k \operatorname{poly}(n)$. All we need to do is brute force over all vectors in the domain of $\{S_i\}_{i>\ell, i\in T}$ and average out $\sigma_k(.)$ of the corresponding sums of rank 1 matrices. □

It follows by Theorem 4.4 and Theorem 4.6 that for any given set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ in isotropic position of squared norm at most $\delta$ we can find a two partitioning $S_1, S_2$ such that $\left\|\sum_{i\in S_j} \mathbf{v}_i \mathbf{v}_i^T\right\| \le 1/2 + O(\sqrt{\delta})$ in time $n^{O(m^{1/3}\delta^{-1/4})}$.

## 4.2 Oracle for ATSP

Next, we construct an oracle for interlacing families related to the asymmetric traveling salesman problem. We say a multivariate polynomial $p \in \mathbb{R}[z_1, \ldots, z_m]$ is real stable if it has no roots in the upper-half complex plane, i.e., $p(\mathbf{z}) \neq 0$ whenever $\operatorname{Im}(z_i) > 0$ for all $i$. The *generating polynomial* of $\mu$ is defined as

$$g_\mu(z_1, \ldots, z_m) = \sum_{S \subseteq [m]} \mu(S) \prod_{i\in S} z_i.$$

We say $\mu$ is a *strongly Rayleigh* probability distribution if $g_\mu(\mathbf{z})$ is real stable. We say $\mu$ is homogeneous if all sets in the support of $\mu$ have the same size. The following theorem is proved by the first and the second authors [AO15b].

**Theorem 4.7** ([AO15b])**.** *Let $\mu$ be a homogeneous strongly Rayleigh probability distribution on subsets of $[m]$ such that for each $i$, $\mathbb{P}[i] < \epsilon_1$. Let $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ be in isotropic position such that for each $i$, $\|\mathbf{v}_i\|^2 \leq \epsilon_2$. Then, there is a set $S$ in the support of $\mu$ such that*

$$\left\| \sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^T \right\| \leq O(\epsilon_1 + \epsilon_2).$$

We give an algorithm that finds such a set $S$ as promised in the above theorem assuming that we have an oracle that for any $\mathbf{z} \in \mathbb{R}^m$ returns $g_\mu(\mathbf{z})$. It follows from the proof of [AO15b] that it is enough to design a $1 + O(\epsilon_1 + \epsilon_2)$-approximation rounding algorithm for the following interlacing family.

For any $1 \leq i \leq m$, let $S_i = \{\mathbf{0}, \mathbf{v}_i\}$. For any $S$ in the support of $\mu$, let $\mathbf{r}_i = \mathbf{v}_i$ if $i \in S$ and $\mathbf{r}_i = \mathbf{0}$ otherwise and we add $\mathbf{r}_1, \ldots, \mathbf{r}_m$ to $\mathcal{F}$. Then, define

$$f_{\mathbf{r}_1,\ldots,\mathbf{r}_m} = \mu(S) \cdot \det \left( xI - \sum_{i=1}^m \mathbf{r}_i \mathbf{r}_i^T \right).$$

It follows from [AO15b] that $\{f_{\mathbf{r}_1,\ldots,\mathbf{r}_m}\}_{\mathbf{r}_1,\ldots,\mathbf{r}_m \in \mathcal{F}}$ is an interlacing family.

Next, we design an algorithm that returns the top $k$ coefficients of any polynomial in this family in time $m^k \operatorname{poly}(n)$.

**Theorem 4.8.** *Given a strongly Rayleigh distribution $\mu$ on subsets of $[m]$ and a set of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$, suppose that we are given an oracle that for any $\mathbf{z} \in \mathbb{R}^m$ returns $g_\mu(\mathbf{z})$. There is an algorithm that for any $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in S_1 \times \cdots \times S_\ell$ with $1 \leq \ell \leq m$ and $1 \leq k \leq n$ returns the top $k$ coefficients of $f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell}$ in time $m^k \operatorname{poly}(n)$.*

*Proof.* Fix $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in S_1 \times \cdots \times S_\ell$ for some $1 \leq \ell \leq m$. First, note that if there is no such element in $\mathcal{F}$, then $g_\mu(1, \ldots, 1, z_{\ell+1}, \ldots, z_m) = 0$ and there is nothing to prove. So assume for some $\mathbf{r}_{\ell+1}, \ldots, \mathbf{r}_m$, $\mathbf{r}_1, \ldots, \mathbf{r}_m \in \mathcal{F}$.

It is sufficient to show that for any $0 \leq k \leq n$, we can compute the coefficient of $x^{n-k}$ of $f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell}$ in time $m^k \operatorname{poly}(n)$. Firstly, observe that since we are only summing up the characteristic polynomials that are consistent with $\mathbf{r}_1, \ldots, \mathbf{r}_\ell$, we can work with the conditional distribution

$$\tilde{\mu} = \{\mu \,|\, i \text{ if } 1 \leq i \leq \ell \text{ and } \mathbf{r}_i = \mathbf{v}_i, \bar{i} \text{ if } 1 \leq i \leq \ell, \mathbf{r}_i = \mathbf{0}\}.$$

Note that since we can efficiently compute $g_\mu(z_1, \ldots, z_m)$, we can also compute $g_{\tilde{\mu}}(z_{\ell+1}, \ldots, z_m)$. For any $i$ that is conditioned to be in, we need to differentiate with respect to $z_i$ and for any $i$ that is conditioned to be out we let $z_i = 0$. Also, note that instead of differentiating we can let $z_i = M$ for a very large number $M$, and then divide the resulting polynomial by $M$. We note that when $\mu$ is a determinantal distribution, which is the case in applications to the asymmetric traveling salesman problem, this differentiation can be computed exactly and efficiently; in other cases, $M$ can be taken to be exponentially large, as we can tolerate an exponentially small error.

Now, we can write

$$f_{\mathbf{r}_1,\ldots,\mathbf{r}_\ell} = \mathbb{E}_{T \sim \tilde{\mu}} \left[ \det \left( xI - \sum_{i \in T} \mathbf{v}_i \mathbf{v}_i^T \right) \right].$$

So, by Proposition 2.1, the coefficient of $x^{n-k}$ in the above is equal to

$$(-1)^k \sum_{T \in \binom{[m]}{k}} \mathbb{P}_{\tilde{\mu}}[T] \cdot \sigma_k \left( \sum_{i \in T} \mathbf{v}_i \mathbf{v}_i^T \right).$$

To compute the above quantity it is enough to brute force over all sets $T \in \binom{[m]}{k}$. For any such $T$ we can compute $\sigma_k(\sum_{i \in T} \mathbf{v}_i \mathbf{v}_i^T)$ in time poly$(n)$. In addition, we can efficiently compute $\mathbb{P}_{\tilde{\mu}}[T]$ using our oracle. It is enough to differentiate with respect to any $i \in T$,

$$\prod_{i \in T : i > \ell} \frac{\partial}{\partial z_i} g_{\tilde{\mu}}(z_{\ell+1}, \dots, z_m)|_{z_{\ell+1}=\dots=z_m=1}.$$

Therefore, the algorithm runs in time $m^k$ poly$(n)$. □

It follows from Theorem 4.4 and Theorem 4.8 that for any homogeneous strongly Rayleigh distribution $\mu$ with marginal probabilities $\epsilon_1$ with an oracle that computes $g_\mu(\mathbf{z})$ for any $\mathbf{z} \in \mathbb{R}^n$, and for any vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ in isotropic position with squared norm at most $\epsilon_2$, we can find a set $S$ in the support of $\mu$ such that $\left\| \sum_{i \in S} \mathbf{v}_i \mathbf{v}_i^T \right\| \le O(\epsilon_1 + \epsilon_2)$ in time $n^{O(m^{1/3}(\epsilon_1+\epsilon_2)^{1/2})}$.

This is enough to get a polyloglog$(m)$ approximation algorithm for asymmetric traveling salesman problem on a graph with $m$ vertices that runs in time $2^{\tilde{O}(m^{1/3})}$ [AO15a].

## References


[AO15a] Nima Anari and Shayan Oveis Gharan. Effective-Resistance-Reducing Flows, Spectrally Thin Trees, and Asymmetric TSP. pages 20–39, 2015. FOCS. 3, 21

[AO15b] Nima Anari and Shayan Oveis Gharan. The kadison-singer problem for strongly rayleigh measures and applications to asymmetric tsp. 2015. 3, 19, 20

[Bec91] József Beck. An algorithmic approach to the Lovász local lemma. I. *Random Structures Algorithms*, 2(4):343–365, 1991. 3

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, 2006. 3

[Coh16] Michael B Cohen. Ramanujan graphs in polynomial time. *arXiv preprint arXiv:1604.03544*, 2016. 3

[CS07] Maria Chudnovsky and Paul Seymour. The roots of the independence polynomial of a clawfree graph. *J. Combin. Theory Ser. B*, 97(3):350–357, 2007. 3

[DP15] Constantinos Daskalakis and Christos Papadimitriou. Sparse covers for sums of indicators. *Probab. Theory Related Fields*, 162(3-4):679–705, 2015. 4

[Erd64] P. Erdős. Extremal problems in graph theory. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36. Publ. House Czechoslovak Acad. Sci., Prague, 1964. 7

[FL06] Shmuel Friedland and Daniel Levy. A polynomial-time approximation algorithm for the number of k-matchings in bipartite graphs. *arXiv preprint cs.CC/0607135*, 28, 2006. 4

[HL72] Ole J. Heilmann and Elliott H. Lieb. Theory of monomer-dimer systems. *Comm. Math. Phys.*, 25:190–232, 1972. 3

[HPS15] Chris Hall, Doron Puder, and William F Sawin. Ramanujan coverings of graphs. *arXiv preprint arXiv:1506.02335*, 2015. 3

[HSV16] Nicholas JA Harvey, Piyush Srivastava, and Jan Vondrák. Computing the independence polynomial in shearer's region for the lll. *arXiv preprint arXiv:1608.02282*, 2016. 3

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, July 2004. 4

[LU95] Felix Lazebnik and Vasiliy A. Ustimenko. Explicit construction of graphs with an arbitrary large girth and of large size. *Discrete Appl. Math.*, 60(1-3):275–284, 1995. ARIDAM VI and VII (New Brunswick, NJ, 1991/1992). 7

[MSS13a] Adam Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing Families I: Bipartite Ramanujan Graphs of All Degrees. In *FOCS*, pages 529–537, 2013. 3, 4, 8, 17

[MSS13b] Adam Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing Families II: Mixed Characteristic Polynomials and the Kadison-Singer Problem. 2013. 3, 4, 5, 17, 18, 19

[MSS15] Adam W. Marcus, Nikhil Srivastava, and Daniel A. Spielman. Interlacing families iv: Bipartite ramanujan graphs of all sizes. to appear in FOCS, 2015. 3

[MT10] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010. 3

[SJ89] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Inf. Comput.*, 82(1):93–133, July 1989. 4

[SS13] Alistair Sinclair and Piyush Srivastava. Lee-Yang theorems and the complexity of computing averages. In *STOC'13—Proceedings of the 2013 ACM Symposium on Theory of Computing*, pages 625–634. ACM, New York, 2013. 3

[Sze39] Gabor Szeg. *Orthogonal polynomials*, volume 23. American Mathematical Soc., 1939. 7

[Val79] L. G. Valiant. The complexity of computing the permanent. *Theoret. Comput. Sci.*, 8(2):189–201, 1979. 3

[Wen91] R. Wenger. Extremal graphs with no $C^4$'s, $C^6$'s, or $C^{10}$'s. *J. Combin. Theory Ser. B*, 52(1):113–116, 1991. 7

[Wil84] James H. Wilkinson. The perfidious polynomial. In *Studies in numerical analysis*, volume 24 of *MAA Stud. Math.*, pages 1–28. Math. Assoc. America, Washington, DC, 1984. 2

# A Appendix A: Proofs of Preliminary Facts

*Proof of Fact 2.4.* If $a = 0$, the conclusion is trivial. Otherwise let $\ell(x) = (x - b)/a$. Then $a\boldsymbol{\mu} + b$ is the vector of the roots of $\chi_{\boldsymbol{\mu}}(\ell(x))$ and similarly $a\boldsymbol{\nu} + b$ is the vector of the roots of $\chi_{\boldsymbol{\nu}}(\ell(x))$. It is easy to see that if $\chi_{\boldsymbol{\mu}}$ and $\chi_{\boldsymbol{\nu}}$ have the same top $k$ coefficients, then after expansion $\chi_{\boldsymbol{\mu}}(\ell(x))$ and $\chi_{\boldsymbol{\nu}}(\ell(x))$ have the same top $k$ coefficients as well, since $\ell(x)$ is linear. □

*Proof of Corollary 2.6.* For $p(x) = x^k$, the statement of the corollary becomes the same as Theorem 2.5. For any other polynomial, we can write

$$p(x) = a_0 + a_1 x + \cdots + a_k x^k.$$

Then we have

$$\sum_{i=1}^{n} p(\mu_i) = \sum_{j=1}^{k} a_j \left( \sum_{i=1}^{n} \mu_i^j \right) = \sum_{j=1}^{k} a_j p_j(\boldsymbol{\mu}).$$

Now because of Theorem 2.5, it follows that

$$\sum_{i=1}^{n} p(\mu_i) = \sum_{j=1}^{k} a_j q_j(e_1(\boldsymbol{\mu}), \ldots, e_k(\boldsymbol{\mu})).$$

The above is a polynomial of $e_1(\boldsymbol{\mu}), \ldots, e_k(\boldsymbol{\mu})$, and it can be computed in time $\mathrm{poly}(k)$, since each $q_j$ can be computed in time $\mathrm{poly}(k)$. □

*Proof of Fact 2.13.* For $x \geq 0$, we have $T_k(1+x) = \cosh(k \cosh^{-1}(1+x))$; here $\cosh^{-1}$ is the inverse of cosh when looked at as a function from $[0, \infty)$ to $[1, \infty)$. Both cosh and $\cosh^{-1}$ are monotonically increasing over the appropriate ranges. Therefore $T_k(1 + x)$ is monotonically increasing for $x \geq 0$.

For $x \geq 0$, we have

$$T_k(1+x) = \cosh(k \cosh^{-1}(1+x)) \geq \frac{\exp(k \cosh^{-1}(1+x))}{2} = \frac{\exp(\cosh^{-1}(1+x))^k}{2} \geq \frac{(1+\sqrt{2x})^k}{2}.$$

In the first inequality we used the fact that $\cosh(x) \geq e^x/2$, and in the last inequality we used the fact that $\exp(\cosh^{-1}(1+x)) \geq 1 + \sqrt{2x}$.

□

*Proof of Fact 2.22.* The maximum eigenvalue of $A$ is characterized by the so called Rayleigh quotient

$$\lambda_{\max}(A) = \max_{\mathbf{x} \in \mathbb{R}^n - \{0\}} \frac{\mathbf{x}^\intercal A \mathbf{x}}{\mathbf{x}^\intercal \mathbf{x}},$$

For $\mathbf{x} = \mathbf{1}$, the all 1s vector, we have

$$\lambda_{\max}(A) \geq \frac{\mathbf{1}^\intercal A \mathbf{1}}{\mathbf{1}^\intercal \mathbf{1}} = \frac{2|E|}{n} = \deg_{\mathrm{avg}}(G).$$

□

# B Appendix B: Constructions of Graphs with Large Girth

In this section we prove Claim 3.10 and Claim 3.11.

*Proof of Claim 3.10.* Assume that $k$ is fixed and Conjecture 2.16 is true for graphs of girth $> 2k$. The graphs promised by Conjecture 2.16 already have average degree $\Omega(n^{1/k})$. We just need to make them bipartite and make sure that their maximum degree is at most $O(n^{1/k})$.

Making the graph bipartite is easy. Given a graph $G$, the following procedure makes it bipartite: Replace each vertex $v$ by $v_1, v_2$. Replace each edge $\{u, v\}$ with two edges $\{v_1, u_2\}$ and $\{u_1, v_2\}$. This procedure doubles the number of edges and the number of vertices, and it is easy to see that it does not decrease the girth.

We can trim the maximum degree by the following procedure: If there is a vertex $v$ where $\deg v > n^{1/k}$, we introduce a new vertex $v'$ and take an arbitrary subset of $n^{1/k}$ edges incident to $v$ and change their endpoint from $v$ to $v'$. By repeating this procedure the graph will eventually have maximum degree bounded by $n^{1/k}$. This procedure increases the number of vertices, but the number of new vertices is easily bounded by $2|E|/n^{1/k}$ because each new vertex has degree $n^{1/k}$ until the end and the number of edges $|E|$ does not change.

Both of the above procedures change the number of vertices. To get graphs of arbitrary size $n$, we can use the following trick: We start with a graph $G_0$ promised by Conjecture 2.16 on $n/c$ vertices, where $c$ is a large constant. We make sure that it has no more than $(n/c)^{1+1/k}$ edges by removing edges if necessary (this can only increase girth). Now we make a bipartite graph $G_1$ from $G_0$ by the aforementioned procedure. From $G_1$ we make $G_2$ using the second procedure which makes sure $\deg_{\max}(G_2) = O(n^{1/k})$. The number of new vertices added by this step is $O(n/c)$. So at the end the total number of vertices in $G_2$ is $O(n/c)$ and if we make $c$ large enough we can make sure it is less than $n$. We can add isolated vertices to $G_2$ until the total number of vertices becomes $n$ and we get $G$. This procedure changes $\deg_{\text{avg}}$ by at most a constant factor and does not change $\deg_{\max}$, so both are $\Theta(n^{1/k})$ at the end. □

*Proof of Claim 3.11.* We build our graphs using Theorem 2.17. Let $t \geq 3$ be the largest odd number such that $2d^t \leq n$. If $n$ is large enough, such a $t$ exists, and we have $t = \Omega(\log n)$. By Theorem 2.17, there exists a $d$-regular bipartite graph $H$ with $2d^t$ vertices and $\text{girth}(H) \geq t + 5 = \Omega(\log n)$. To construct $G$, put as many copies of $H$ as possible side by side making sure the number of vertices does not grow more than $n$. At the end add some isolated vertices to make the total number of vertices $n$ and get $G$. It is easy to see that number of isolated vertices added at the end is at most $n/2$. These vertices have degree 0 and the rest have degree $d$. Therefore $\deg_{\text{avg}}(G) \geq d/2$ and $\deg_{\max}(G) = d$. □