Hermitian Laplacians and a Cheeger Inequality for the Max-2-Lin Problem

Huan Li
School of Computer Science, Fudan University, China
huanli16@fudan.edu.cn

He Sun
School of Informatics, University of Edinburgh, United Kingdom
h.sun@ed.ac.uk

Luca Zanetti
Department of Computer Science and Technology, University of Cambridge, United Kingdom
luca.zanetti@cl.cam.ac.uk

Abstract

We study spectral approaches for the MAX-2-LIN(k) problem, in which we are given a system of m linear equations of the form $x_i - x_j \equiv c_{ij} \bmod k$, and required to find an assignment to the n variables $\{x_i\}$ that maximises the total number of satisfied equations.

We consider Hermitian Laplacians related to this problem, and prove a Cheeger inequality that relates the smallest eigenvalue of a Hermitian Laplacian to the maximum number of satisfied equations of a MAX-2-LIN(k) instance $\mathcal{I}$. We develop an $\widetilde{O}(kn^2)$ time algorithm that, for any $(1-\varepsilon)$-satisfiable instance, produces an assignment satisfying a $\left(1 - O(k)\sqrt{\varepsilon}\right)$-fraction of equations. We also present a subquadratic-time algorithm that, when the graph associated with $\mathcal{I}$ is an expander, produces an assignment satisfying a $\left(1 - O(k^2)\varepsilon\right)$-fraction of the equations. Our Cheeger inequality and first algorithm can be seen as generalisations of the Cheeger inequality and algorithm for MAX-CUT developed by Trevisan.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Approximation algorithms analysis; Mathematics of computing → Spectra of graphs

Keywords and phrases Spectral methods, Hermitian Laplacians, the Max-2-Lin problem, Unique Games

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.71

Related Version A full version of the paper is available at `https://arxiv.org/abs/1811.10909`.

Funding *Luca Zanetti*: Supported by the ERC Starting Grant "DYNAMIC MARCH".

Acknowledgements We are very grateful to Mihai Cucuringu for valuable discussions on Hermitian Laplacian matrices and their applications. We would also like to thank Chris Heunen for some fruitful conversations on topics closely related to this paper.

1 Introduction

In the MAX-2-LIN(k) problem, we are given a system of m linear equations of the form $u_i - v_i \equiv c_i \bmod k$, where $u_i, v_i \in \{x_1, \ldots, x_n\}$ and each equation has weight b_i. The objective is to find an assignment to the variables x_i that maximises the total weight of satisfied equations. As an important case of Unique Games [8, 15], the MAX-2-LIN(k) problem has been extensively studied in theoretical computer science. This problem is known to be NP-hard to approximate within a ratio of $11/12 + \delta$ for any constant $\delta > 0$ [9, 13], and it is conjectured to be hard to distinguish between MAX-2-LIN(k) instances for which a $(1-\varepsilon)$-fraction of equations can be satisfied versus instances for which only an ε-fraction can

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 71; pp. 71:1–71:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

be satisfied [16]. On the algorithmic side, there has been a number of LP and SDP-based algorithms proposed for the MAX-2-LIN(k) problem (e.g., [6, 12, 15, 26]), and the case of $k = 2$, which corresponds to the classical MAX-CUT problem for undirected graphs [10, 14], has been widely studied over the past fifty years.

In this paper we investigate efficient spectral algorithms for MAX-2-LIN(k). For any MAX-2-LIN(k) instance $\mathcal{I}$ with n variables, we express $\mathcal{I}$ by a Hermitian Laplacian matrix $L_{\mathcal{I}} \in \mathbb{C}^{n\times n}$, and analyse the spectral properties of $L_{\mathcal{I}}$. In comparison to the well-known Laplacian matrix for undirected graphs [7], complex-valued entries in $L_{\mathcal{I}}$ are able to express directed edges in the graph associated with $\mathcal{I}$, and at the same time ensure that all the eigenvalues of $L_{\mathcal{I}}$ are real-valued. We demonstrate the power of our Hermitian Laplacian matrices by relating the maximum number of satisfied equations of $\mathcal{I}$ to the spectral properties of $L_{\mathcal{I}}$. In particular, we develop a Cheeger inequality that relates partial assignments of $\mathcal{I}$ to $\lambda_1(L_{\mathcal{I}})$, the smallest eigenvalue of $L_{\mathcal{I}}$. Based on a recursive application of the algorithm behind our Cheeger inequality, as well as a spectral sparsification procedure for MAX-2-LIN(k) instances, we present an approximation algorithm for MAX-2-LIN(k) that runs in $\widetilde{O}(k \cdot n^2)$ time[1] . To the best of our knowledge, this is the first purely spectral polynomial-time algorithm for the MAX-2-LIN(k) problem with approximation guarantees that matches SDP-based ones for constant values of k. The formal statement of our result is as follows:

▶ **Theorem 1.** *There is an $\widetilde{O}(k \cdot n^2)$-time algorithm such that, for any given MAX-2-LIN(k) instance $\mathcal{I}$ with optimum $1 - \varepsilon$, the algorithm returns an assignment ϕ satisfying at least a $(1 - O(k)\sqrt{\varepsilon})$-fraction of the equations*[2].

Our result can be viewed as a generalisation of the MAX-CUT algorithm by Trevisan [27], who derived a Cheeger inequality that relates the value of the maximum cut to the smallest eigenvalue of an undirected graph's adjacency matrix. The proof of Trevisan's Cheeger inequality, however, is based on constructing sweep sets in $\mathbb{R}$, while in our setting constructing sweep sets in $\mathbb{C}$ is needed, as the underlying graph defined by $L_{\mathcal{I}}$ is directed and eigenvectors of $L_{\mathcal{I}}$ are in $\mathbb{C}^n$. The other difference between our result and the one in [27] is that the goal of the MAX-CUT problem is to find a *bipartition* of the vertex set, while for the MAX-2-LIN(k) problem we need to use an eigenvector to find k vertex-disjoint subsets, which corresponds to subsets of variables assigned to the same value.

Our approach also shares some similarities with the one by Goemans and Williamson [11], who presented a 0.793733-approximation algorithm for MAX-2-LIN(3) based on Complex Semidefinite Programming. The objective function of their SDP relaxation is, in fact, exactly the quadratic form of our Hermitian Laplacian matrix $L_{\mathcal{I}}$, although this matrix was not explicitly defined in their paper. In addition, their rounding scheme divides the complex unit ball into k regions according to the angle with a random vector, which is part of our rounding scheme as well. Therefore, if one views Trevisan's work [27] as a spectral analogue to the celebrated SDP-based algorithm for MAX-CUT by Goemans and Williamson [10], our result can be seen as a spectral analogue to the Goemans and Williamson's algorithm for MAX-2-LIN(k).

We further prove that, when the undirected graph associated with a MAX-2-LIN(k) instance is an expander, the approximation ratio from Theorem 1 can be improved. Our result is formally stated as follows:

---

[1] The notation $\widetilde{O}(\cdot)$ suppresses poly-logarithmic factors in n, m, and k.

[2] An instance $\mathcal{I}$ has optimum $1 - \varepsilon$, if the maximum fraction of the total weights of satisfied equations is $1 - \varepsilon$.

▶ **Theorem 2.** *Let $\mathcal{I}$ be an instance of MAX-2-LIN(k) on a d-regular graph with n vertices and suppose its optimum is $1-\varepsilon$. There is an $\widetilde{O}\left(nd + \frac{n^{1.5}}{k\sqrt{\varepsilon}}\right)$-time algorithm that returns an assignment $\phi : V \to [k]$ satisfying at least a*

$$1 - O(k^2) \cdot \frac{\varepsilon}{\lambda_2^3(\mathcal{L}_{\mathcal{U}})} \quad (1)$$

fraction of equations in $\mathcal{I}$, where $\lambda_2(\mathcal{L}_{\mathcal{U}})$ is the second smallest eigenvalue of the normalised Laplacian matrix of the underlying undirected graph $\mathcal{U}$.

Our technique is similar to the one by Kolla [18], which was used to show that solving the MAX-2-LIN(k) problem on expander graphs is easier. In [18], a MAX-2-LIN(k) instance is represented by the label-extended graph, and the algorithm is based on an exhaustive search in a subspace spanned by eigenvectors associated with eigenvalues close to 0. When the underlying graph of the MAX-2-LIN(k) instance has good expansion, this subspace is of dimension k. Therefore, the exhaustive search runs in time $O\left(2^k + \mathrm{poly}(n \cdot k)\right)$, which is polynomial-time when $k = O(\log n)$. Comparing with the work in [18], we show that, when the underlying graph has good expansion, the eigenvector associated with the smallest eigenvalue $\lambda_1(\mathcal{L}_{\mathcal{I}})$ of the Hermitian Laplacians suffices to give a good approximation. We notice that Arora et al. [4] already showed that, for expander graphs, it is possible to satisfy a $1 - O(\varepsilon \log(1/\varepsilon))$ fraction of equations in polynomial time without any dependency on k. Their algorithm is based on an SDP relaxation.

1.1 Other related work

There are many research results for the MAX-2-LIN(k) problem (e.g., [6, 12, 15, 26]), and we briefly discuss the ones most closely related to our work. For the MAX-2-LIN(k) problem and Unique Games, spectral techniques are often employed to analyse the Laplacian matrix of the Label-Extended graph (see, e.g., the aforementioned [18]), which has a strong connection with our Hermitian Laplacian: the latter can be seen as one of the blocks that arise in a particular block-diagonalisation of the former. Arora et al. [3], instead, use spectral techniques to obtain a particular decomposition of the constraint graph of a Unique Games instance, and exploit this decomposition to design an $\exp\left((kn)^{O(\varepsilon)}\right)\mathrm{poly}(n)$-time algorithm for Unique Games. Regarding polynomial-time algorithms, Charikar et al. [6] propose an SDP-based algorithm for Unique Games that satisfies a $1 - O(\sqrt{\varepsilon \log k})$ fraction of constraints, which is nearly optimal assuming the Unique Games Conjecture [16]. We remark that canonical SDP programs for Unique Games can be solved in nearly-linear time [25].

Our result also relates to the research on spectral methods for synchronisation problems. For example, the adjacency matrix corresponding to our Hermitian Laplacian is considered by Singer [23] in relation to an angular synchronisation problem. The relation between the eigenvectors of such matrix and the MAX-2-LIN(k) problem is also mentioned but without offering formal approximation guarantees. Bandeira et al. [5] prove a Cheeger-type inequality that relates the spectra of an operator, the graph connection Laplacian, to "how well" an instance of the $O(d)$-synchronisation problem can be solved. Their results, however, are not directly comparable to ours: even though our Hermitian Laplacian can also be seen as a graph connection Laplacian for an $SO(2)$-synchronisation problem, our goal is to assign the n vertices to at most k elements of $SO(2)$, while the goal of Bandeira et al. is to assign each vertex to a possibly different element of $O(d)$.

2 Hermitian Matrices for MAX-2-LIN(k)

We can write an instance of MAX-2-LIN(k) by $\mathcal{I} = (G, k)$, where $G = (V, E, b, c)$ denotes a directed graph with an edge weight function $b : E \to \mathbb{R}^+$ and an edge color function $c : E \to [k]$, where $[k] \stackrel{\text{def}}{=} \{0, 1, \ldots, k-1\}$. More precisely, every equation $u_i - v_i \equiv c_i \bmod k$ with weight b_i corresponds to a directed edge (u_i, v_i) with weight $b(u_i, v_i) = b_{u_i v_i} = b_i$ and color $c(u_i, v_i) = c_{u_i v_i} = c_i$. In the rest of this paper, we will assume that G is weakly connected, and write $u \rightsquigarrow v$ if there is a directed edge from u to v. The conjugate transpose of any vector $x \in \mathbb{C}^n$ is denoted by x^*.

We define the Hermitian adjacency matrix $A_{\mathcal{I}} \in \mathbb{C}^{n \times n}$ for instance $\mathcal{I}$ by

$$(A_{\mathcal{I}})_{uv} \stackrel{\text{def}}{=} \begin{cases} b_{uv}\omega_k^{c_{uv}} & u \rightsquigarrow v, \\ b_{vu}\overline{\omega_k}^{\,c_{vu}} & v \rightsquigarrow u, \\ 0 & \text{otherwise,} \end{cases}$$

where $\omega_k = \exp\left(\frac{2\pi i}{k}\right)$ is the complex k-th root of unity, and $\overline{\omega_k} = \exp\left(-\frac{2\pi i}{k}\right)$ is its conjugate. We define the degree-diagonal matrix $D_{\mathcal{I}}$ by $(D_{\mathcal{I}})_{uu} = d_u$ where d_u is the weighted degree given by $d_u \stackrel{\text{def}}{=} \sum_{u \rightsquigarrow v} b_{uv} + \sum_{v \rightsquigarrow u} b_{vu}$. The Hermitian Laplacian matrix is then defined by $L_{\mathcal{I}} = D_{\mathcal{I}} - A_{\mathcal{I}}$, and the corresponding normalised Laplacian matrix by $\mathcal{L}_{\mathcal{I}} = D_{\mathcal{I}}^{-1/2} L_{\mathcal{I}} D_{\mathcal{I}}^{-1/2} = I - D_{\mathcal{I}}^{-1/2} A_{\mathcal{I}} D_{\mathcal{I}}^{-1/2}$. The eigenvalues of any matrix A are expressed by $\lambda_1(A) \leq \ldots \leq \lambda_n(A)$. The quadratic forms of $L_{\mathcal{I}}$ can be related to the corresponding instance of MAX-2-LIN(k) by the following lemma.

▶ **Lemma 3.** *For any vector $x \in \mathbb{C}^n$, we have $x^* L_{\mathcal{I}} x = \sum_{u \rightsquigarrow v} b_{uv} \left\| x_u - \omega_k^{c_{uv}} x_v \right\|^2$ and*

$$x^* L_{\mathcal{I}} x = 2 \sum_{u \in V} d_u \left\| x_u \right\|^2 - \sum_{u \rightsquigarrow v} b_{uv} \left\| x_u + \omega_k^{c_{uv}} x_v \right\|^2 .$$

The lemma below presents a qualitative relationship between the eigenvector associated with $\lambda_1(\mathcal{L}_{\mathcal{I}})$ and an assignment of $\mathcal{I}$.

▶ **Lemma 4.** *All eigenvalues of $\mathcal{L}_{\mathcal{I}}$ are in the range $[0, 2]$. Moreover, $\lambda_1(\mathcal{L}_{\mathcal{I}}) = 0$ if and only if there exists an assignment satisfying all equations in $\mathcal{I}$.*

3 A Cheeger inequality for $\lambda_1(\mathcal{L}_{\mathcal{I}})$ and MAX-2-LIN(k)

The discrete Cheeger inequality [1] shows that, for any undirected graph G, the conductance h_G of $G = (V, E)$ can be approximated by the second smallest eigenvalue of G's normalised Laplacian matrix $\mathcal{L}_G$, i.e.,

$$\frac{\lambda_2(\mathcal{L}_G)}{2} \leq h_G \leq \sqrt{2 \cdot \lambda_2(\mathcal{L}_G)}. \tag{2}$$

Moreover, the proof of the second inequality above is constructive, and indicates that a subset $S \subset V$ with conductance at most $\sqrt{2 \cdot \lambda_2(\mathcal{L}_G)}$ can be found by using the second bottom eigenvector of $\mathcal{L}_G$ to embed vertices on the real line. As one of the most fundamental results in spectral graph theory, the Cheeger inequality has found applications in the study of a wide range of optimisation problems, e.g., graph partitioning [20], max-cut [27], and many practical problems like image segmentation [22] and web search [17].

In this section, we develop connections between $\lambda_1(\mathcal{L}_{\mathcal{I}})$ and MAX-2-LIN(k) by proving a Cheeger-type inequality. Let $\phi : \{x_1, \ldots, x_n\} \to [k] \cup \{\perp\}$ be an arbitrary *partial assignment* of an instance $\mathcal{I}$, where $\phi(x_i) = \perp$ means that the assignment of x_i has not been decided.

These variables' assignments will be determined through some recursive construction, which will be elaborated in Section 5. We remark that this framework of recursively computing a partial assignment was first introduced by Trevisan [27], and our theorem can be viewed as a generalisation of the one in [27], which corresponds to the $k = 2$ case of ours.

To relate quadratic forms of $\mathcal{L}_G$ with the objective function of the MAX-2-LIN(k) problem, we introduce a *penalty* function as follows:

▶ **Definition 5.** *Given a partial assignment $\phi : \{x_1, \ldots, x_n\} \to [k] \cup \{\perp\}$ and a directed edge (u, v), the penalty of (u, v) with respect to ϕ is defined by*

$$p_{uv}^{\phi}(\mathcal{I}) \stackrel{\text{def}}{=} \begin{cases} 0 & \phi(u) \neq \perp, \phi(v) \neq \perp, \phi(u) - \phi(v) \equiv c_{uv} \mod k \\ 1 & \phi(u) \neq \perp, \phi(v) \neq \perp, \phi(u) - \phi(v) \not\equiv c_{uv} \mod k \\ 0 & \phi(u) = \phi(v) = \perp \\ 1 - \frac{1}{k} & \text{exactly one of } \phi(u), \phi(v) \text{ is } \perp. \end{cases} \tag{3}$$

For simplicity, we write p_{uv}^{ϕ} when the underlying instance $\mathcal{I}$ is clear from the context.

The values of p_{uv}^{ϕ} from Definition 5 are chosen according to the following facts: (1) If both u and v's values are assigned, then their penalty is 1 if the equation $\phi(u) - \phi(v) \not\equiv c_{uv} \mod k$ associated with (u, v) is unsatisfied, and 0 otherwise; (2) If both u and v's values are $\perp$, then their penalty is temporally set to 0. Their penalty will be computed when u and v's assignment are determined during a later recursive stage. (3) If exactly one of u, v is assigned, p_{uv}^{ϕ} is set to $1 - 1/k$, since a random assignment to the other variable makes the edge (u, v) satisfied with probability $1/k$, hence p_{uv}^{ϕ} is set to $1 - 1/k$.

Without loss of generality, we only consider ϕ for which $\phi(u) \neq \perp$ for at least one vertex u, and define the penalty of assignment ϕ by

$$p^{\phi} \stackrel{\text{def}}{=} \frac{2 \sum_{u \rightsquigarrow v} b_{uv} p_{uv}^{\phi}}{\mathsf{Vol}(\phi)}, \tag{4}$$

where $\mathsf{Vol}(\phi) \stackrel{\text{def}}{=} \sum_{\phi(u) \neq \perp} d_u$. Notice that the p_{uv}^{ϕ}'s value is multiplied by b_{uv} in accordance with the objective of MAX-2-LIN(k) that maximises the total weight of satisfied assignments. Also, we multiply p_{uv}^{ϕ} by 2 in the numerator since edges with at least one assigned endpoint are counted at most twice in $\mathsf{Vol}(\phi)$. Notice that, as long as G is weakly connected, $p^{\phi} = 0$ if and only if all edges are satisfied by ϕ and, in general, the smaller the value of p^{ϕ}, the more edges are satisfied by ϕ. With this in mind, we define the *imperfectness* $p(\mathcal{I})$ of $\mathcal{I}$ to quantify how close $\mathcal{I}$ is to an instance where all equations can be satisfied by a single assignment.

▶ **Definition 6.** *Given any MAX-2-LIN(k) instance $\mathcal{I} = (G, k)$, the imperfectness of $\mathcal{I}$ is defined by $p(\mathcal{I}) \stackrel{\text{def}}{=} \min_{\phi \in ([k] \cup \{\perp\})^V \setminus \{\perp\}^V} p^{\phi}$.*

The main result of this section is a Cheeger-type inequality that relates $p(\mathcal{I})$ and $\lambda_1(\mathcal{L}_{\mathcal{I}})$, which is summarised in Theorem 7. Note that, since $\sin(x) \geq (2/\pi) \cdot x$ for $x \in [0, \pi/2]$, the factor before $\sqrt{2\lambda_1}$ in the theorem statement is at most $(2 + k/4)$ for $k \geq 2$.

▶ **Theorem 7.** *Let λ_1 be the smallest eigenvalue of $\mathcal{L}_{\mathcal{I}}$. It holds that*

$$\frac{\lambda_1}{2} \leq p(\mathcal{I}) \leq \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right) \sqrt{2\lambda_1}. \tag{5}$$

Moreover, given the eigenvector associated with λ_1, there is an $O(m + n \log n)$-time algorithm that returns a partial assignment ϕ such that

$$\frac{\lambda_1}{2} \leq p^{\phi} \leq \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right) \sqrt{2\lambda_1}. \tag{6}$$

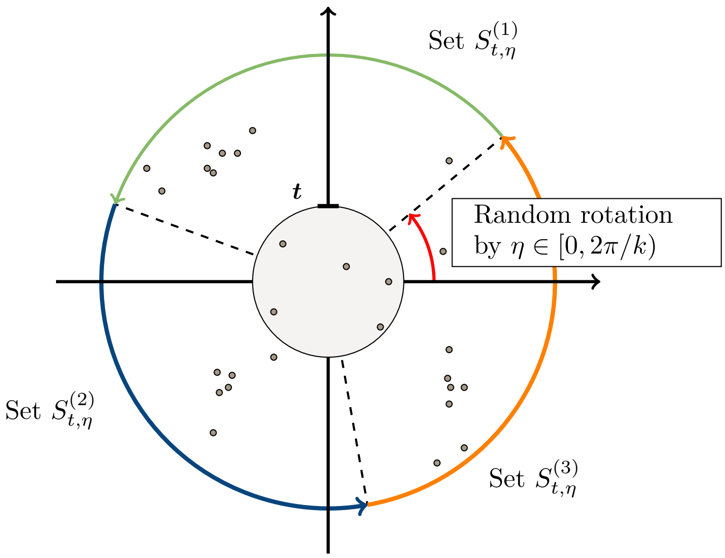

Figure 1 Illustration of the proof for Theorem 7 for the case of $k = 3$. The gray circle is obtained by sweeping $t \in [0, 1]$, and the red arrow represents a random angle $\eta \in [0, 2\pi/k)$. A partial assignment is determined by the values of η and t.

Proof Sketch of Theorem 7. We present an overview of the proof here, and a complete proof of the theorem can be found in the full version of the paper. The easy direction of (5), i.e., $\lambda_1/2 \le p(\mathcal{I})$, follows from the Courant-Fischer characterisation of eigenvalues and that the eigenvector problem is a relaxation of MAX-2-LIN(k). Hence, we will mainly sketch the techniques used to prove the other direction of (5). We assume that $z \in \mathbb{C}^n$ is the vector such that

$$\frac{z^* L_{\mathcal{I}} z}{z^* D_{\mathcal{I}} z} = \lambda_1,$$

and prove the existence of an assignment ϕ based on z satisfying

$$p^{\phi} \le \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\sqrt{2\lambda_1} = \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\sqrt{2 \cdot \frac{z^* L_{\mathcal{I}} z}{z^* D_{\mathcal{I}} z}}.$$

We first scale each coordinate of z such that $\max_{u \in V} \|z_u\|^2 = 1$. In this way z can be seen as an embedding of the vertices to the complex unit ball. For any real numbers $t \ge 0$ and $\eta \in [0, \frac{2\pi}{k})$, we define k sets of vertices indexed by $j \in [k]$ as follows:

$$S^{(j)}_{t,\eta} = \left\{ u \;\middle|\; \|z_u\| \ge t \text{ and } \theta(z_u, e^{i\eta}) \in \left[j \cdot \frac{2\pi}{k}, (j+1) \cdot \frac{2\pi}{k} \right) \right\}.$$

Here, we use $\theta(a, b) \in [-\pi, \pi)$ to represent the angle from $b \in \mathbb{C}$ to $a \in \mathbb{C}$, i.e., $\frac{a}{\|a\|} = \frac{b}{\|b\|} \exp(i\theta(a, b))$. We then define an assignment $\phi_{t,\eta}$ where $\phi_{t,\eta}(u) = j$ if there is $j \in [k]$ such that $u \in S^{(j)}_{t,\eta}$, and $\phi_{t,\eta}(u) = \perp$ otherwise. By definition, the k vertex sets correspond to the vectors in the k regions of the unit ball after each vector is rotated by η radians counterclockwise. The role of t is to only consider the coordinates z_u with $\|z_u\| \ge t$. This is illustrated in Figure 1.

Now we assume $t \in [0,1]$ is chosen such that t^2 follows from a uniform distribution over $[0,1]$, and η is chosen uniformly at random from $[0, 2\pi/k)$. Further calculations show that

$$\mathbb{E}_{t,\eta}\left[\mathsf{Vol}(\phi_{t,\eta})\right] = \sum_{u\in V} d_u \cdot \mathbb{P}\left[\|z_u\| \geq t\right] = \sum_{u \in V} d_u \left\|z_u\right\|^2 = z^* D_{\mathcal{I}} z,$$

and

$$\mathbb{E}_{t,\eta}\left[2\sum_{u\rightsquigarrow v} b_{uv}\, p^{\phi}_{uv}\right] \leq \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\cdot\sqrt{z^* L_{\mathcal{I}} z}\cdot\sqrt{2 z^* D_{\mathcal{I}} z}.$$

Hence, it holds that

$$\frac{\mathbb{E}_{t,\eta}\left[2\sum_{u\rightsquigarrow v} b_{uv} p^{\phi}_{uv}\right]}{\mathbb{E}_{t,\eta}\left[\mathsf{Vol}(\phi_{t,\eta})\right]} \leq \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\cdot\sqrt{2\cdot\frac{z^* L_{\mathcal{I}} z}{z^* D_{\mathcal{I}} z}}.$$

This implies by linearity of expectation that

$$\mathbb{E}_{t,\eta}\left[2\sum_{u\rightsquigarrow v} b_{uv} p^{\phi}_{uv} - \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\cdot\mathsf{Vol}(\phi_{t,\eta})\cdot\sqrt{2\cdot\frac{z^* L_{\mathcal{I}} z}{z^* D_{\mathcal{I}} z}}\right] \leq 0,$$

and existence of an assignment ϕ satisfying (6).

Now we turn to the runtime needed to find such a vertex set. Notice that we need to find t and η such that $\phi_{t,\eta}$ satisfies (6). Therefore, we construct two sequences of *sweep sets*: the first is based on t, and the second is based on η. For constructing the sweep sets based on t, the algorithm increases t from 0 to 1, and updates the corresponding conditional expectation looking only at the edges incident with u whenever t exceeds $\|z_u\|$. Notice that each edge (u, v) will be updated at most twice, i.e., the step when t reaches $\|z_u\|$ and the step when t reaches $\|z_v\|$, and the total runtime needed to update $\mathsf{Vol}(\phi_{t,\eta})$ is $O(m)$. Hence, the total runtime for constructing the sweep sets based on t is $O(m)$. The runtime analysis for constructing the sweep sets based on η is similar: the algorithm increases η from 0 to $2\pi/k$, and updates the penalties p^{ϕ}_{uv} of the edges (u, v) only if the assignment of u or v changes. Since every edge will be updated at most twice, the runtime for constructing the sweep sets based on η is $O(m)$ as well. Hence, the total runtime of the algorithm is $O(m + n \log n)$. ◀

▶ Remark 8. We remark that the factors $\lambda_1/2$ and $\sqrt{\lambda_1}$ in Theorem 7 are both tight within constant factors. The tightness can be derived directly from Section 5 of [27], since when $k = 2$, our inequality is the same as the one in [27] up to constant factors.

We also remark that the factor of k in Theorem 7 is necessary, which is shown by the following instance: the linear system has nk variables where every variable belongs to one of k sets $S_0, \ldots, S_{k-1}$ with $|S_i| = n$ for any $0 \leq i \leq k-1$. Now, for any i, we add n equations of the form $x_u - x_v = 1 \mod k$ with $x_u \in S_i$, $x_v \in S_j$, and $j = i + 1 \mod k$, and n equations of the form $x_u - x_v = 1 \mod k$ with $x_u \in S_i$, $x_v \in S_j$, and $j = i + 2 \mod k$. This instance is constructed such that the underlying graph is regular, and every assignment could only satisfy at most half of the equations, implying that the imperfectness is $p(\mathcal{I}) = \Omega(1)$. However, mapping each variable in S_i to the root of unity ω_k^i, it's easy to see that $\lambda_i(\mathcal{L}_{\mathcal{I}}) = O(1/k)$. Hence Theorem 7 is tight with respect to k.

Finally, we compare the proof techniques of Theorem 7 with other Cheeger-type inequalities in the literature: first of all, most of the Cheeger-type inequalities (e.g., [1, 19, 20, 27]) consider the case where every eigenvector is in $\mathbb{R}^n$ and are only applicable for undirected graphs, while for our problem the graph G associated with $\mathcal{I}$ is directed and eigenvectors of

$A_{\mathcal{I}}$ are in $\mathbb{C}^n$. Therefore, constructing sweep sets in $\mathbb{C}$ is needed, which is more involved than proving similar Cheeger-type inequalities (e.g., [1, 27]). Secondly, by dividing the complex unit ball into k regions, we are able to show that a partial assignment corresponding to k disjoint subsets can be found using a single eigenvector. This is quite different from the techniques used for finding k vertex-disjoint subsets of low conductance in an undirected graph, where k eigenvectors are usually needed (e.g. [19, 20, 21]).

We also remark that, while sweeping through values of t is needed to obtain *any* guarantee on the penalty of the partial assignment computed, we could in principle just choose a random angle η: in this way, however, the partial assignment returned would satisfy (6) only in expectation.

4 Sparsification for MAX-2-LIN(k)

We have seen in Section 3 that, given any vector in $\mathbb{C}^n$ whose quadratic form with $L_{\mathcal{I}}$ is close to $\lambda_1(L_{\mathcal{I}})$, we can compute a partial assignment of $\mathcal{I}$ with bounded approximation guarantee. In Section 5 we will show that a total assignment can be found by recursively applying this procedure on variables for which an assignment has not yet been fixed. In particular, we will show that every iteration takes a time nearly-linear in the number of equations of our instance, which can be quadratic in the number of variables. To speed-up each iteration and obtain a time per iteration that is nearly-linear in the number of variables, we need to sparsify our input instance $\mathcal{I}$.

In this section we show that the construction of spectral sparsifiers by effective resistance sampling introduced in [24] can be generalised to sparsify MAX-2-LIN(k) instances. In particular, given an instance $\mathcal{I}$ of MAX-2-LIN(k) with n variables and m equations, we can find in nearly-linear time a sparsified instance $\mathcal{J}$ with about $nk\log(nk)$ equations such that for any partial assignment $\phi : V \to [k]$, the number of unsatisfied equations in $\mathcal{J}$ is preserved within a constant factor. This means that we can apply our algorithm for MAX-2-LIN(k) to a sparsified instance $\mathcal{J}$, and any dependency on m in our runtime can be replaced by $nk\log(nk)$. We remark that we could simply apply uniform sampling to obtain a sparsified instance. However, this would in the end result in an additive error in the fraction of unsatisfied equations, much like in the case of the original Trevisan's result for MAX-CUT [27]. With our construction, instead, we only lose a small multiplicative error.

To construct a sparsified instance $\mathcal{J}$, we introduce label-extended graphs and their Laplacian matrices to characterise the original MAX-2-LIN(k) instance. Let $P \in \mathbb{R}^{k\times k}$ be the permutation matrix where $P_{ij} = 1$ if $i \equiv j + 1 \bmod k$, and $P_{ij} = 0$ otherwise. We define the adjacency matrix $\widetilde{A}_{\mathcal{I}} \in \left(\mathbb{R}^{k\times k}\right)^{n\times n}$ for the label-extended graph of instance $\mathcal{I}$, where each entry of $\widetilde{A}_{\mathcal{I}}$ is a matrix in $\mathbb{R}^{k\times k}$ given by

$$(\widetilde{A}_{\mathcal{I}})_{uv} \stackrel{\text{def}}{=} \begin{cases} b_{uv} P^{c_{uv}} & u \rightsquigarrow v, \\ b_{vu} \left(P^{\intercal}\right)^{c_{vu}} & v \rightsquigarrow u, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We then define the degree-diagonal matrix $\widetilde{D}_{\mathcal{I}} \in \left(\mathbb{R}^{k\times k}\right)^{n\times n}$ by $(\widetilde{D}_{\mathcal{I}})_{uu} = d_u \cdot I_{k\times k}$, where $I_{k\times k}$ is the $k \times k$ identity matrix, and define the Laplacian matrix by $\widetilde{L}_{\mathcal{I}} = \widetilde{D}_{\mathcal{I}} - \widetilde{A}_{\mathcal{I}}$. Notice that the Hermitian Laplacian $\mathcal{L}_{\mathcal{I}}$ is a *compression* of $\widetilde{L}_{\mathcal{I}}$, i.e., there exists an orthogonal projection U such that $U^*\widetilde{L}_{\mathcal{I}}U = \mathcal{L}_{\mathcal{I}}$.

For any assignment $\phi : V \to [k]$, we construct an indicator vector $\widetilde{x}_{\mathcal{I}} \in \left(\mathbb{R}^k\right)^n$ by $\left(\widetilde{x}_{\mathcal{I}}\right)_u = e_{\phi(u)}$, where $e_j \in \mathbb{R}^k$ is the j-th standard basis vector. Then, it is easy to show that the total weight of unsatisfied equations for ϕ is $(1/2) \cdot \widetilde{x}_{\mathcal{I}}^{\intercal} \widetilde{L}_{\mathcal{I}} \widetilde{x}_{\mathcal{I}}$.[3]

We show that, for every unsatisfiable instance $\mathcal{I}$, there is a sparsified MAX-2-LIN(k) instance $\mathcal{J}$ such that the quadratic forms between $\widetilde{L}_{\mathcal{I}}$ and $\widetilde{L}_{\mathcal{J}}$ are approximately preserved. This implies that, when looking at the same assignment, the total weights of unsatisfied equations in $\mathcal{I}$ and $\mathcal{J}$ are approximately preserved. Notice that we can decide whether there is an assignment satisfying all the equations in $\mathcal{I}$ by fixing the assignment of an arbitrary vertex and determining assignments for other vertices accordingly, and therefore we only need to consider the case when $\mathcal{I}$ is unsatisfiable. The main result of the section is as follows:

▶ **Theorem 9.** *There is an algorithm that, given an unsatisfiable instance $\mathcal{I}$ of MAX-2-LIN(k) with n variables and m equations and parameter $0 < \delta < 1$, returns in $\widetilde{O}(mk)$ time an instance $\mathcal{J}$ with the same set of variables and $O\left((1/\delta^2) \cdot nk \log(nk)\right)$ equations. Furthermore, with high probability it holds for any vector $x \in \left(\mathbb{R}^k\right)^n$ that $(1-\delta) x^{\intercal} \widetilde{L}_{\mathcal{I}} x \le x^{\intercal} \widetilde{L}_{\mathcal{J}} x \le (1+\delta) x^{\intercal} \widetilde{L}_{\mathcal{I}} x$.*

5 Algorithm for MAX-2-LIN(k)

Theorem 9 tells us that, given an instance $\mathcal{I}^*$, we can find a sparse instance $\mathcal{I}$ so that the quadratic forms of the corresponding Laplacians $\mathcal{L}_{\mathcal{I}^*}$ and $\mathcal{L}_{\mathcal{I}}$ are approximately the same. Therefore throughout this section we assume that the input instance $\mathcal{I}$ for MAX-2-LIN(k) with n variables has $m = \widetilde{O}\left((1/\delta^2) \cdot nk\right)$ equations for some parameter $\delta > 0$. Recall that Theorem 7 shows that, for any MAX-2-LIN(k) instance $\mathcal{I}$, given an eigenvector for the smallest eigenvalue $\lambda_1(\mathcal{L}_{\mathcal{I}})$, we can obtain a partial assignment ϕ satisfying

$$p^{\phi} \le \left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right) \sqrt{2\lambda_1}. \tag{8}$$

Now we show that, by a repeated application of Theorem 7 on the subset of the equations of $\mathcal{I}$ for which both variables are unassigned, we can obtain a full assignment of $\mathcal{I}$. Our algorithm closely follows the one by Trevisan [27] and is described in Algorithm 1.

To achieve the guarantees of (8), however, we would need to compute the eigenvector corresponding to $\lambda_1(\mathcal{L}_{\mathcal{I}})$ *exactly*. To obtain a nearly-linear time algorithm, instead, we relax this requirement and compute a vector z that well-approximates this eigenvector. In particular, the following lemma shows that, for any δ, we can compute a vector $z \in \mathbb{C}^n$ satisfying (9) in nearly-linear time.

▶ **Lemma 10.** *For any given error parameter δ, there is an $\tilde{O}\left(\left(1/\delta^3\right) \cdot kn\right)$ time algorithm that returns $z \in \mathbb{C}^n$ satisfying (9).*

To analyse Algorithm 1, we introduce some notation. Let t be the number of recursive executions of Algorithm 1. For any $1 \le j \le t+1$, let $\mathcal{I}_j$ be the instance of MAX-2-LIN(k) in the j-th execution. We indicate with $\rho_j m$ the number of equations in $\mathcal{I}_j$, where $0 \le \rho_j \le 1$. Notice that $\mathcal{I}_1 = \mathcal{I}$ and $\mathcal{I}_{t+1} = \emptyset$. We assume that the maximum number of equations in $\mathcal{I}_j$ that can be satisfied by an assignment is $(1 - \varepsilon_j)\rho_j m$, with $\varepsilon = \varepsilon_1$. Also notice that it holds for any $1 \le j \le t$ that $\varepsilon_j \rho_j m \le \varepsilon m$, which implies $\varepsilon_j \le \varepsilon/\rho_j$. The next theorem presents the performance of our algorithm, whose informal version is Theorem 1.

[3] We remark that, if we use the Hermitian Laplacian matrices $L_{\mathcal{I}}$ directly instead, this relation only holds up to an $O(k)$ factor. That is why we sparsify the matrix $\widetilde{L}_{\mathcal{I}}$ instead.

Algorithm 1 RECURSIVECONSTRUCT($\mathcal{I}, \delta$).

Compute vector $z \in \mathbb{C}^n$ satisfying

$$\frac{z^* L_{\mathcal{I}} z}{z^* D_{\mathcal{I}} z} \leq (1+2\delta)\lambda_1(\mathcal{L}_{\mathcal{I}}); \tag{9}$$

Apply the algorithm from Theorem 7 to compute $\phi : V \to [k] \cup \{\perp\}$ such that

$$p^\phi \leq (1+\delta)\left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right)\sqrt{2\lambda_1}; \tag{10}$$

if $2p^\phi \geq (1-1/k)\,\mathsf{Vol}(\phi)$ **then**
    return random full assignment $\phi' : V \to [k]$;
    ▷ the case where the current assignment is worse than a random assignment
else if ϕ is a full assignment (i.e. $\phi(V) \subseteq [k]$) **then**
    return ϕ;
    ▷ The recursion terminates if every variable's assignment is determined
else
    $\mathcal{I}' \leftarrow$ set of equations from $\mathcal{I}$ in which both variables' assignments are not determined;
    if $\mathcal{I}' = \emptyset$ **then**
        set $\phi(u)$ to be an arbitrary assignment if $\phi(u) = \perp$ for any u;
        return ϕ;
    else
        $\phi_1 \leftarrow$ RECURSIVECONSTRUCT($\mathcal{I}', \delta$);
        return $\phi \cup \phi_1$;

▶ **Theorem 11.** *Given an instance $\mathcal{I}$ of MAX-2-LIN(k) whose optimum is $1-\varepsilon$ and a parameter $\delta > 0$, the algorithm* RECURSIVECONSTRUCT($\mathcal{I}, \delta$) *returns in $\widetilde{O}\left((1/\delta^3) \cdot kn^2\right)$ time an assignment ϕ satisfying at least $1 - 8\nu\sqrt{\varepsilon}$ fraction of the equations, where*

$$\nu \stackrel{\text{def}}{=} (1+\delta)\left(2 - \frac{2}{k} + \frac{1}{2\sin(\pi/k)}\right) = O(k).$$

The following corollary which states how much our algorithm beats a random assignment follows from Theorem 1.

▶ **Corollary 12.** *Given a MAX-2-LIN(k) instance $\mathcal{I}$ whose optimum is ξ and a constant $\delta > 0$, Algorithm 1 returns in $\tilde{O}\left((1/\delta^3) \cdot n^2 k\right)$ time an assignment ϕ satisfying at least $(1/k + \tau)\,\xi$ fraction of the equations, where $\tau = \Omega\left(\frac{1}{k^3}\right)$.*

6 Algorithm for MAX-2-LIN(k) on expanders

In this section we further develop techniques for analysing Hermitian Laplacian matrices by presenting a subquadratic-time approximation algorithm for the MAX-2-LIN(k) problem on expander graphs. Our proof technique is inspired by Kolla's algorithm [18]. However, in contrast to the algorithm in [18], we use the Hermitian Laplacian to represent a MAX-2-LIN(k) instance and show that, when the underlying graph has good expansion, a good approximate solution is encoded in the eigenvector associated with $\lambda_1(\mathcal{L}_{\mathcal{I}})$. We assume that G is a d-regular graph, and hence $\mathcal{I} = (G, k)$ is a MAX-2-LIN(k) instance with n variables and $nd/2$

equations whose optimum is $1-\varepsilon$. One can view $\mathcal{I}$ as an instance generated by modifying ε fraction of the constraints (i.e., edges) from a completely satisfiable instance $\widehat{\mathcal{I}} = (\widehat{G}, k)$. Hence, a satisfiable assignment $\psi : V \to [k]$ for $\widehat{\mathcal{I}}$ will satisfy at least a $(1-\varepsilon)$-fraction of equations in $\mathcal{I}$.

Now we discuss the techniques used to prove Theorem 2. Let $y_\psi \in \mathbb{C}^n$ be the normalised "indicator vector" of ψ, i.e., $(y_\psi)_u = \frac{1}{\sqrt{n}}\omega_k^{\psi(u)}$. Then it holds that

$$(y_\psi)^* \mathcal{L}_{\widehat{\mathcal{I}}} y_\psi = \frac{1}{d}\sum_{u \rightsquigarrow v} b_{uv} \left\|(y_\psi)_u - \omega_k^{c_{uv}}(y_\psi)_v\right\|^2 = 0.$$

Hence y_ψ is an eigenvector associated with $\lambda_1\left(\mathcal{L}_{\widehat{\mathcal{I}}}\right) = 0$. We denote by $\mathcal{U}$ the underlying undirected graph of G, and denote by $\mathcal{L}_\mathcal{U}$ the normalised Laplacian of $\mathcal{U}$. Note that since $\mathcal{U}$ is undirected, $\mathcal{L}_\mathcal{U}$ only contains real-valued entries. We first show that the eigenvalues of $\mathcal{L}_{\widehat{\mathcal{I}}}$, the normalised Laplacian of the completely satisfiable instance, and of $\mathcal{L}_\mathcal{U}$, the normalised Laplacian of the underlining undirected graph $\mathcal{U}$, coincide. Since $\mathcal{L}_\mathcal{U}$ is the Laplacian matrix of an expander graph, this implies that there is a gap between $\lambda_1\left(\mathcal{L}_{\widehat{\mathcal{I}}}\right)$ and $\lambda_2\left(\mathcal{L}_{\widehat{\mathcal{I}}}\right)$.

▶ **Lemma 13.** *It holds for all $1 \le i \le n$ that $\lambda_i\left(\mathcal{L}_{\widehat{\mathcal{I}}}\right) = \lambda_i\left(\mathcal{L}_\mathcal{U}\right)$.*

Next we bound the perturbation of the bottom eigenspace of $\mathcal{L}_{\widehat{\mathcal{I}}}$ when the latter is turned into $\mathcal{L}_\mathcal{I}$. In particular, Lemma 14 below proves that this perturbation does not affect too much to the vectors that have norm spreads out uniformly over all their coordinates.

▶ **Lemma 14.** *Let $f \in \mathbb{C}^n$ be a vector such that $\|f_u\| = \frac{1}{\sqrt{n}}$ for all $u \in V$. It holds that*

$$\left\|\left(\mathcal{L}_\mathcal{I} - \mathcal{L}_{\widehat{\mathcal{I}}}\right) f\right\| \le 2\sqrt{\varepsilon}. \tag{11}$$

Based on Lemma 14, we prove that the change from $\mathcal{L}_{\widehat{\mathcal{I}}}$ to $\mathcal{L}_\mathcal{I}$ doesn't have too much influence on the eigenvector associated with $\lambda_1(\mathcal{L}_\mathcal{I})$. For simplicity, let $\lambda_2 = \lambda_2(\mathcal{L}_{\widehat{\mathcal{I}}}) = \lambda_2(\mathcal{L}_\mathcal{U})$.

▶ **Lemma 15.** *Let $f_1 \in \mathbb{C}^n$ be a unit eigenvector associated with $\lambda_1(\mathcal{L}_\mathcal{I})$. Then we have $\left\|\left(\mathcal{L}_\mathcal{I} - \mathcal{L}_{\widehat{\mathcal{I}}}\right) f_1\right\| \le 20\sqrt{\varepsilon/\lambda_2}$.*

We then prove the following lemma which shows that the eigenvector f_1 corresponding to $\lambda_1(\mathcal{L}_\mathcal{I})$ is close to y_ψ, the indicator vector of the optimal assignment ψ.

▶ **Lemma 16.** *Let $f_1 \in \mathbb{C}^n$ be a unit eigenvector associated with $\lambda_1(\mathcal{L}_\mathcal{I})$. Then, there exist $\alpha, \beta \in \mathbb{C}$ and a unit vector $y_\perp \in \mathbb{C}^n$ orthogonal to y_ψ (i.e. $(y_\perp)^* y_\psi = 0$) such that $f_1 = \alpha y_\psi + \beta y_\perp$ and $\|\beta\| \le 30\sqrt{\varepsilon/\lambda_2^3}$.*

Based on Lemma 16, f_1 is close to the indicator vector of an optimal assignment rotated by some angle. In particular, we have that

$$\left\|f_1 - \frac{\alpha}{\|\alpha\|} y_\psi\right\| = \sqrt{(1-\|\alpha\|)^2 + \|\beta\|^2} \le \sqrt{1-\|\alpha\|^2 + \|\beta\|^2} = \sqrt{2}\,\|\beta\| \le 30\sqrt{\frac{2\varepsilon}{\lambda_2^3}}, \tag{12}$$

where $\frac{\alpha}{\|\alpha\|} y_\psi$ is the vector that encodes the information of an assignment that satisfies all the equations in $\widehat{\mathcal{I}}$ and at least $1-\varepsilon$ fraction of equations in $\mathcal{I}$. Therefore, our goal is to recover $\frac{\alpha}{\|\alpha\|} y_\psi$ from f_1.

Proof of Theorem 2. Let ψ be the optimal assignment of $\mathcal{I}$ satisfying $1-\varepsilon$ fraction of equations, which is also a completely satisfying assignment of $\widehat{\mathcal{I}}$. Let f_1 be a unit eigenvector associated with $\lambda_1(\mathcal{L}_\mathcal{I})$. By Lemma 16, there exists $\alpha, \beta \in \mathbb{C}$ such that $f_1 = \alpha y_\psi + \beta y_\perp$ where $\|\beta\| \leq 30\sqrt{\varepsilon/\lambda_2^3}$. Our goal is to find a vector $z_\phi \in \mathbb{C}^n$, which equals the indicator vector of ϕ ratoted by some angle and satisfies

$$\|f_1 - z_\phi\| \leq \left\| f_1 - \frac{\alpha}{\|\alpha\|} y_\psi \right\| \leq 30\sqrt{\frac{2\varepsilon}{\lambda_2^3}}, \tag{13}$$

where the last inequality follows by (12). The assignment ϕ corresponding to such a z_ϕ will give us that the fraction of unsatisfied equations by ϕ is

$$\begin{aligned}
p^\phi(\mathcal{I}) &\leq 10k^2 z_\phi^* \mathcal{L}_\mathcal{I} z_\phi \\
&= 10k^2 (z_\phi - f_1 + f_1)^* \mathcal{L}_\mathcal{I} (z_\phi - f_1 + f_1) \\
&\leq k^2 \left((z_\phi - f_1)^* \mathcal{L}_\mathcal{I} (z_\phi - f_1) + f_1^* \mathcal{L}_\mathcal{I} f_1 + 2\left\| (z_\phi - f_1)^* \mathcal{L}_\mathcal{I} f_1 \right\| \right) \\
&\leq 10k^2 \left(2\left\| z_\phi - f_1 \right\|^2 + \lambda_1(\mathcal{L}_\mathcal{I}) + 2\left\| z_\phi - f_1 \right\| \sqrt{\lambda_1(\mathcal{L}_\mathcal{I})} \right) \\
&\leq 10k^2 \left(2 \cdot 900 \cdot \frac{2\varepsilon}{\lambda_2^3} + 2\varepsilon + 2 \cdot 30 \cdot \sqrt{\frac{2\varepsilon}{\lambda_2^3}} \cdot \sqrt{2\varepsilon} \right) \\
&\leq 100000k^2 \cdot \frac{\varepsilon}{\lambda_2^3},
\end{aligned}$$

where the factor $10k^2$ above follows from the fact that $\left\| 1 - \omega_k^j \right\|^2$ is at least $1/(10k^2)$ for $j = 1, \ldots, k-1$.

To find such vector z_ϕ satisfying (13), we define $\phi_\eta(u) = \arg\min_{j \in [k]} \left\| (f_1)_u - \mathrm{e}^{\eta i} \omega_k^j \right\|$. Notice that, since $\frac{\alpha}{\|\alpha\|}$ is equal to $\mathrm{e}^{\eta i}$ for some $\eta \in [0, 2\pi)$, by defining $(z_{\phi_\eta})_u = \mathrm{e}^{\eta i} \omega_k^{\phi_\eta(u)}$ the solution to the following optimisation problem $\min_{\eta \in [0,2\pi)} \left\| z_{\phi_\eta} - f_1 \right\|$ gives us a vector that satisfies (13). To solve this optimisation problem, we notice that it suffices to consider η in the range $[0, 2\pi/k)$. Therefore, we simply enumerate all η's over the following discrete set:

$$\left\{ \frac{t\sqrt{\varepsilon}}{\sqrt{n}} \;\middle|\; t = 0, 1, \ldots, \left\lceil \frac{2\pi\sqrt{n}}{k\sqrt{\varepsilon}} \right\rceil \right\}.$$

By enumerating this set, we can find an assignment ϕ and an η such that

$$\left\| f_1 - z_{\phi_\eta} \right\| \leq \left\| f_1 - \frac{\alpha}{\|\alpha\|} y_\psi \right\| + O(\sqrt{\varepsilon}),$$

which is enough to get our desired approximation. Since the size of this set is $O\left(\frac{\sqrt{n}}{k\sqrt{\varepsilon}}\right)$, the total running time is $O\left(\frac{n^{1.5}}{k\sqrt{\varepsilon}}\right)$ plus the running time needed to compute f_1. ◀

7 Concluding remarks

Our work leaves several open questions for further research: while the factor of k in our Cheeger inequality (Theorem 7) is needed, it would be interesting to see if it's possible to construct a different Laplacian for which a similar Cheeger inequality holds with a smaller dependency on k. For example, instead of embedding vertices in $\mathbb{C}$ and mapping assignments to roots of unity, one could consider embedding vertices in higher dimensions using the bottom k eigenvectors of the Laplacian of the label extended graph, and see if a relation between the imperfectness ratio of Definition 6 and the k-th smallest eigenvalue of this Laplacian still holds.

Finally, we observe that several cut problems in directed graphs can be formulated as special cases of MAX-2-LIN(k) (see, e.g., [2, 11]). Because of this, we believe the Hermitian Laplacians studied in our paper will have further applications in the development of fast algorithms for combinatorial problems on directed graphs, and might have further connections to Unique Games.

References

1 Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

2 Gunnar Andersson, Lars Engebretsen, and Johan Håstad. A New Way of Using Semidefinite Programming with Applications to Linear Equations mod p. *Journal of Algorithms*, 39(2):162–204, 2001.

3 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential Algorithms for Unique Games and Related Problems. *Journal of the ACM*, 62(5), 2015.

4 Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy. In *40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 21–28, 2008.

5 Afonso S. Bandeira, Amit Singer, and Daniel A. Spielman. A Cheeger inequality for the graph connection Laplacian. *SIAM J. Matrix Anal. Appl.*, 34(4):1611–1630, 2013.

6 Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *38th Annual ACM Symposium on Theory of Computing (STOC'06)*, pages 205–214, 2006.

7 Fan R. K. Chung. Spectral Graph Theory. *Regional Conference Series in Mathematics, American Mathematical Society*, 92:1–212, 1997.

8 Uriel Feige and László Lovász. Two-prover one-round proof systems: Their power and their problems. In *24th Annual ACM Symposium on Theory of Computing (STOC'92)*, pages 733–744, 1992.

9 Uriel Feige and Daniel Reichman. On systems of linear equations with two variables per equation. In *7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'04)*, pages 117–127, 2004.

10 Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

11 Michel X. Goemans and David P. Williamson. Approximation algorithms for Max-3-Cut and other problems via complex semidefinite programming. *Journal of Computer and System Sciences*, 68(2):442–470, 2004.

12 Anupam Gupta and Kunal Talwar. Approximating unique games. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, pages 99–106, 2006.

13 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

14 Richard M. Karp. Reducibility Among Combinatorial Problems. In *a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.

15 Subhash Khot. On the power of unique 2-prover 1-round games. In *34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 767–775, 2002.

16 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

17 Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604–632, 1999.

18 Alexandra Kolla. Spectral algorithms for unique games. *Computational Complexity*, 20(2):177–206, 2011.

19 Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, Shayan Oveis Gharan, and Luca Trevisan. Improved Cheeger's inequality: analysis of spectral partitioning algorithms through higher order spectral gap. In *45th Annual ACM Symposium on Theory of Computing (STOC'13)*, pages 11–20, 2013.

20 James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. *Journal of the ACM*, 61(6):37:1–37:30, 2014.

21 Richard Peng, He Sun, and Luca Zanetti. Partitioning Well-Clustered Graphs: Spectral Clustering Works! *SIAM Journal on Computing*, 46(2):710–743, 2017.

22 Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

23 Amit Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20, 2011.

24 Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

25 David Steurer. Fast SDP Algorithms for Constraint Satisfaction Problems. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 684–697, 2010.

26 Luca Trevisan. Approximation algorithms for unique games. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 197–205, 2005.

27 Luca Trevisan. Max Cut and the Smallest Eigenvalue. *SIAM Journal on Computing*, 41(6):1769–1786, 2012.