Spectrally Robust Graph Isomorphism

Alexandra Kolla[1]

Department of Computer Science, University of Colorado at Boulder
alexandra.kolla@colorado.edu

Ioannis Koutis[2]

Department of Computer Science, New Jersey Institute of Technology
ioannis.koutis@njit.edu

Vivek Madan

Department of Computer Science, University of Illinois, Urbana-Champaign
vmadan2@illinois.edu

Ali Kemal Sinop

TOBB University of Economics and Technology, Ankara, Turkey
asinop@gmail.com

──── **Abstract** ────

We initiate the study of spectral generalizations of the graph isomorphism problem.

(a) The *Spectral Graph Dominance (SGD)* problem:
On input of two graphs G and H does there exist a permutation π such that $G \preceq \pi(H)$?

(b) The *Spectrally Robust Graph Isomorphism (κ-SRGI)* problem:
On input of two graphs G and H, find the smallest number κ over all permutations π such that $\pi(H) \preceq G \preceq \kappa c \pi(H)$ for some c. SRGI is a natural formulation of the network alignment problem that has various applications, most notably in computational biology.

$G \preceq cH$ means that for all vectors x we have $x^T L_G x \leq c x^T L_H x$, where L_G is the Laplacian G.

We prove NP-hardness for SGD. We also present a κ^3-approximation algorithm for SRGI for the case when both G and H are bounded-degree trees. The algorithm runs in polynomial time when κ is a constant.

1 Introduction

Network alignment, a problem loosely defined as the comparison of graphs under permutations, has a very long history of applications in disparate fields [5]. Notably, alignment of protein and other biological networks are among the most recent and popular applications [18, 6]. There are several heuristic algorithms for the problem; naturally some of them are based on generalizations of the graph isomorphism problem, mostly including variants of the robust

---

graph isomorphism problem which asks for a permutation that minimizes the number of 'mismatched' edges [16].

Robust graph isomorphism may not be always an appropriate problem for applications where one wants to certify the 'functional' equivalence of two graphs. Consider for example the case when G and H are two random constant-degree expanders. While they can be arguably functionally equivalent (e.g. as information dispersers), all permutations will incur a large number of edge mismatches, deeming the two graphs very unsimilar. Functional equivalence is of course an application-dependent notion. In the case of protein networks, it is understood that proteins act as electron carriers [9]. Thus it is reasonable to model them as electrical resistive networks that are algebraically captured by graph Laplacian matrices [4]. Going back to the graph isomorphism problem, we note the simple fact that the Laplacian matrices of two isomorphic graphs share the same eigenvalues, with the corresponding eigenspaces being identical up to the isomorphism. We can them aim for a **spectrally robust** version of graph isomorphism (SRGI) which allows for similar eigenvalues and approximately aligned eigenspaces, up to a permutation.

In lieu of using directly the eigenvalues and eigenspaces to define SRGI, we will rely on the much cleaner notion of spectral graph similarity, which underlies spectral sparsification of graphs, a notion that has been proven extremely fruitful in algorithm design [2, 14]. More concretely, let us introduce the precise notion of similarity we will be using.

▶ **Definition 1** (dominance). We say that graph G dominates graph H ($G \preceq H$), when for all vectors x, we have $x^T L_G x \le x^T L_H x$, where L_G is the standard Laplacian matrix for G.

▶ **Definition 2** (κ-similarity). We say that graphs G and H are κ-similar, when when there exist numbers β and γ, such that $\kappa = \gamma/\beta$ and $\beta H \preceq G \le \gamma H$.[3]

We are now ready to introduce our main problem.

SPECTRALLY ROBUST GRAPH ISOMORPHISM (κ-SRGI): Given two graphs G, H, does there exist a permutation π on $V(G)$ such that G and $\pi(H)$ are κ-similar?

It can been shown that this definition does imply approximately equal eigenvalues and aligned eigenspaces [13], thus testing for κ-similarity under permutations is indeed a spectrally robust version of graph isomorphism. Going back to our example with the two random expanders, it is well-understood that G and $\pi(H)$ will be κ-similar for a constant κ and for all permutations π, which is what we intuitively expect.

We view spectrally robust graph isomorphism as an interesting theoretical problem due to its close relationship with other fundamental algorithmic questions. In particular, it can be easily seen that κ-SRGI is equivalent to the graph isomorphism problem when $\kappa = 1$. As we will discuss in more detail, SRGI can also be viewed as a natural generalization of the minimum distortion problem [10]. Up to our knowledge, the spectral-similarity approach to network alignment has been mentioned earlier only in [20]. In view of the vast number of works on GI ([11, 7, 21, 19, 3, 15, 1] to mention a few) as well as the works on the robust graph isomorphism problem [16] and the minimum distortion problem [10], we find it surprising that SRGI has not received a wider attention.

The goal of this work is to prove some initial results on SRGI and stimulate further research. Towards that end, we provide the first algorithm for this problem, for the case when both graphs are trees.

---

[3] Graphs are weighted and cG is graph G with its edge weights multiplied by c.

▶ **Theorem 3.** *Given two κ-similar trees G and H of maximum degree d, there exists an algorithm running in time $O(n^{O(k^2 d)})$ which finds a mapping certifying that the they are at most κ^4-similar.*

The algorithm for trees is already highly involved, which gives grounds for speculating that the problem is NP-hard. We give evidence that this may be indeed true by turning our attention to the one-sided version of the problem.

SPECTRAL GRAPH DOMINANCE (SGD): Given two graphs G, H, does there exist a permutation π such that G dominates $\pi(H)$?

Given two graphs G and H that have the same eigenvalues, it is not hard to prove that if G and H are not isomorphic, then G cannot dominate H (and vice-versa). Combining this with the fact that isomorphic graphs have the same eigenvalues, we infer that SGD is at least graph isomorphism-hard. The second contribution of this work is the following theorem.

▶ **Theorem 4.** *The* SPECTRAL GRAPH DOMINANCE *problem is NP-hard.*

Theorem 4 is proved in Section 2. We can actually prove a slightly stronger theorem that restricts one of the input graphs to be a tree.

1.1   Related Work

The Robust Graph Isomorphism problem (RGI) asks for a permutation that minimizes the number of mismatched edges. O'Donnell *et al.* [16] gave a constant factor hardness for RGI . The Minimum Distortion problem (MD) views graphs as distance metrics, using the shortest path metric. The goal is to find a mapping between the two metrics so as to minimize the maximum distortion. The connection between SRGI and MD stems from the observation that if two tree graphs G and H are κ-similar up to a permutation π, then the distortion between the induced graph distances of G and $\pi(H)$ is at most κ. For the MD problem, Kenyon *et al.* [10] gave an algorithm which finds a solution with distortion at most α (provided that it exists) in time $\mathrm{poly}(n) \exp(d^{O(\alpha^3)})$, for a tree of degree at most g and an arbitrary weighted graph. They also prove that this problem is NP-hard to approximate within a constant factor.

The term 'spectral alignment' has been used before in [6] in the context of spectral relaxation of the graph matching function. The algorithm in [18] is more spectral 'in spirit' because it uses directly the spectral of the normalized Laplacians of several subgraphs to construct complicated 'graph signatures' that are then compared for similarity. There is no underlying objective function that drives the computation of these signatures, but we imagine that the proposed algorithm or some variant of it, may be a reasonably good practical candidate for SRGI. The work by Tsourakakis [20] proposes an algorithm that searches for the optimal permutation via a sequence of transpositions; however the running time of the algorithm does not have any non-trivial sub-exponential upper bound.

2   Graph Dominance

Preliminaries.   Given a weighted graph $G = (V, E, w)$ we denote by E_G its edges. The Laplacian L_G of G is the matrix defined by $L(i, j) = -w_{ij}$ and $L(i, i) = \sum_{i \neq j} w_{ij}$. The quadratic form $\mathcal{R}(G, x)$ of G is the function defined as:

$$\mathcal{R}(G, x) = x^T L_G x = \sum_{i,j} w_{ij}(x_i - x_j)^2. \tag{1}$$

Let G^∞ be the infinite graph with vertex set equal to all points on the plane with integer coordinates. There is an edge between two points of G^∞ if they have Euclidean distance one. A *cubic subgrid* is a finite subgraph of G^∞ such that all of its nodes have degree at most 3.

The main ingredient of the proof is the following theorem.

▶ **Theorem 5.** *Let G be a cubic subgrid and C be the cycle graph, both on n vertices. There exists a permutation π such that $\pi(C) \preceq G$ if and only if G contains a Hamiltonian cycle.*

Proof. If G contains a Hamiltonian cycle $\pi(C)$, then equation 1 directly implies that $\pi(C) \preceq G$. To prove the converse assume that G does not contain a Hamiltonian cycle and let H be a permutation of C such that $|E_G \cap E_H|$ is maximized. We prove a number of claims and lemmas.

▶ **Claim 1.** *Let G', H' be the graphs obtained by deleting the common edges between G and H respectively. Then, $\mathcal{R}(G, x) < \mathcal{R}(H, x)$ if and only if $\mathcal{R}(G', x) < \mathcal{R}(H', x)$.*

Proof. Let F be the graph induced by the edges shared by G and H. By equation 1 we have $\mathcal{R}(G, x) = \mathcal{R}(G', x) + \mathcal{R}(F, x)$ and $\mathcal{R}(H, x) = \mathcal{R}(H', x) + \mathcal{R}(F, x)$. The claim follows.   ◀

▶ **Claim 2.** *Let v be a vertex with $deg_{G'}(v) = 1$, $deg_{H'}(v) = 0$ and let G'' be the graph obtained from G' after deleting the edge incident to v, and set $H'' = H'$. Then, there exists a vector x s.t. $\mathcal{R}(H', x) > \mathcal{R}(G', x)$ iff there exists a vector y s.t. $\mathcal{R}(H'', y) > \mathcal{R}(G'', y)$.*

Proof. Let x be a vector such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$. Since G'' is a subgraph of G', we have $\mathcal{R}(G'', x) \leq \mathcal{R}(G', x) < \mathcal{R}(H', x) = \mathcal{R}(H'', x)$, and we can take $y = x$. For the converse, assume that there is a vector y such that $\mathcal{R}(H'', y) > \mathcal{R}(G'', y)$. Let (v, w) be the edge incident to v in G'. We define a vector x as follows: $x_u = y_u$ for all $u \neq v$ and $x_v = y_w$. Since, $deg_{H''}(v) = 0$, we have $\mathcal{R}(H'', y) = \mathcal{R}(H'', x)$. On the other hand, G' and G'' agree on all the edges except (v, w). Hence, $\mathcal{R}(G', x) = \mathcal{R}(G'', x) + (x_v - x_w)^2 = \mathcal{R}(G'', x)$. The two vectors x and y agree on all the entries except at v, and the degree of v in G'' is zero. Hence, $\mathcal{R}(G'', x) = \mathcal{R}(G'', y)$. Combining all the inequalities, we get:
$\mathcal{R}(G', x) = \mathcal{R}(G'', x) = \mathcal{R}(G'', x) < \mathcal{R}(H'', y) = \mathcal{R}(H'', x) = \mathcal{R}(H', x)$.   ◀

▶ **Claim 3.** *Let G' and H' be the graphs obtained by deleting the shared edges between G and H as in Claim 1. Let $\tilde{G}$ and $\tilde{H}$ be the graphs obtained starting from G' and H' and repeatedly applying the edge deletion operation of Claim 2. Then, for any vertex u, $deg_{\tilde{G}}(u) \leq deg_{\tilde{H}}(u) + 1$.*

Proof. Since G is a cubic subgrid graph and H is a cycle, $deg_G(u) \leq 3$, $deg_H(u) = 2$, for all vertices u. Deleting edges shared between G and H decreases the degree of any given vertex by the same amount in G and H. Moreover, at any given step, we only delete edges from G'. Hence, $deg_{\tilde{G}}(u) \leq deg_{\tilde{H}}(u) + 1$.   ◀

▶ **Claim 4.** *Let G' and H' be the graphs obtained by deleting the shared edges between G and H as in Claim 1. If there exists a vertex v such that $deg_{G'}(v) = 1, deg_{H'}(v) \geq 1$. Then, there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.*

Proof. Let the edge incident to v in G' be (v, w) and an edge incident to v in H' be (v, u). Since, H' and G' do not share any edge, we have $u \neq w$. Let $x \in \mathbb{R}^n$ be a vector defined as follows: $x_v = 0, x_w = \frac{1}{2}$ and $x_t = 1$ otherwise. We have $\mathcal{R}(H', x) > (x_v - x_u)^2 = 1$, and

$$\mathcal{R}(G', x) = (x_v - x_w)^2 + \sum_{(w,a) \in E'_G, a \neq v} (x_w - x_a)^2$$

Vertex w has at most two neighbors other than v in G, since $deg_{G'}(v) \leq 3$ and for any such neighbor a, we have $(x_a - x_w)^2 = (\frac{1}{2} - 1)^2 = 1/4$. Hence $\mathcal{R}(G', x) \leq 3/4 < 1 \leq \mathcal{R}(H', x)$.   ◄

Let G' and H' be the graphs obtained by deleting the shared edges between G and H as in Claim 1. Claims 3 and 4 allow us to assume without loss of generality that there is no degree one vertex in G' and for all vertices u, $deg_{G'}(u) \leq deg_{H'}(u) + 1$. For convenience, we will refer to the edges of G' as **black edges** and edges of H' as **blue edges**.

▶ **Lemma 6.** *If there exist five vertices u, v, w_1, w_2, w_3 such that*
- $(u, w_1), (w_1, w_2), (w_2, w_3)$ *are black edges and* $(v, w_1), (v, w_2)$ *are* ***not*** *black edges.*
- (u, v) *is a blue edge and* (u, w_2) *is* ***not*** *a blue edge.*

Then, there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

Proof. Let x be the vector with $x_u = 0$, $x_v = 2$, $x_{w_1} = \frac{1}{3}$ and $x_{w_2} = \frac{2}{3}$, and $x_t = 1$ otherwise. We have

$$
\begin{aligned}
\mathcal{R}(H', x) &\geq (x_u - x_v)^2 + \sum_{\substack{(u,a) \in E_{H'} \\ a \neq v}} (x_u - x_a)^2 + \sum_{\substack{(v,b) \in E_{H'} \\ b \neq u}} (x_v - x_b)^2 \\
&= 4 + (deg_{H'}(u) - 1) \cdot (0 - 1)^2 + (deg_{H'}(v) - 1) \cdot (2 - 1)^2 \\
&= deg_{H'}(u) + deg_{H'}(v) + 2 \geq deg_{G'}(u) + deg_{G'}(v) \qquad \text{(Claim 3)}
\end{aligned}
$$

and

$$
\begin{aligned}
\mathcal{R}(G', x) &= (x_u - x_{w_1})^2 + (x_{w_1} - x_{w_2})^2 + (x_{w_2} - x_{w_3})^2 + \sum_{\substack{(u,a) \in E_{G'} \\ a \neq w_1}} (x_u - x_a)^2 \\
&\quad + \sum_{\substack{(w_1,c) \in E_{G'} \\ c \neq w_2, u}} (x_{w_1} - x_c)^2 + \sum_{\substack{(v,b) \in E_{G'}}} (x_v - x_b)^2 + \sum_{\substack{(w_2,d) \in E_{G'} \\ d \neq w_1, w_3}} (x_{w_2} - x_d)^2.
\end{aligned}
$$

We observe that **(i)** The first three terms are equal to $\frac{1}{9}$. **(ii)** There is at most one edge (w_1, c) for $c \neq w_2, u$. Also, since w_1 is not incident to v, we have $x_c = 1$. Thus the fifth term is at most $\frac{4}{9}$. **(iii)** There is at most one edge (w_2, d) for $d \neq w_1, w_3$. In addition, G' is a subgrid, so there is no cycle of length 3 and w_2 is not incident to u. Also, w_2 is not incident to v, by assumption. So, it must be that $x_d = 1$ and the last term is at most equal to $\frac{1}{9}$. **(iv)** Since G' and H' do not share an edge, u is not connected to v. By assumption u is also not incident to w_2. So, it must be that $x_a = 1$ and the fourth term is equal to $deg_{G'}(u) - 1$. **(v)** Vertex v is not connected to u, w_1, w_2. Thus it must be $x_b = 1$ and the sixth term is equal to $deg_{G'}(v)$. Collecting the terms gives $\mathcal{R}(G, x) \leq deg_{G'}(u) + deg_{G'}(v) - \frac{1}{9}$ and the Lemma follows.   ◄

▶ **Lemma 7.** *If there exist four different vertices u, v, w_1, w_2 such that*
- w_1 *has only two black adjacent edges (u, w_1) and (w_1, w_2)*
- (u, v) *is a blue edge.*

Then, there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

Proof. Let x be a vector with $x_u = 0, x_v = 2, x_{w_1} = \frac{1}{2}$ and $x_t = 1$ otherwise. We have

$$
\begin{aligned}
\mathcal{R}(H', x) &\geq (x_u - x_v)^2 + \sum_{\substack{(u,a) \in E_{H'} \\ a \neq v}} (x_u - x_a)^2 + \sum_{\substack{(v,b) \in E_{H'} \\ b \neq u}} (x_v - x_b)^2 \\
&= 4 + (deg_{H'}(u) - 1) \cdot (0 - 1)^2 + (deg_{H'}(v) - 1) \cdot (2 - 1)^2 \quad \text{(no shared edges)} \\
&= deg_{H'}(u) + deg_{H'}(v) + 2 \geq deg_{G'}(u) + deg_{G'}(v) \qquad \text{(Claim 3)}
\end{aligned}
$$

and

$$\mathcal{R}(G', x) = (x_u - x_{w_1})^2 + (x_{w_1} - x_{w_2})^2 + \sum_{\substack{(u,a) \in E_{G'} \\ a \neq w_1}} (x_u - x_a)^2 + \sum_{(v,b) \in E_{G'}} (x_v - x_b)^2$$

Since G' and H' do not share an edge, $(x_u - x_a)^2$ and $(x_v - x_b)^2$ terms are $(0-1)^2$ and $(2-1)^2$ respectively. We have

$$\mathcal{R}(G', x) = \frac{1}{4} + \frac{1}{4} + (deg_{G'}(u) - 1) \cdot 1 + deg_{G'}(v) \cdot 1 = deg_{G'}(u) + deg_{G'}(v) - \frac{1}{2}.$$

The Lemma follows. ◀

▶ **Lemma 8.** *If there exists a degree three vertex in G', then there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.*

Proof. Since $deg_{G'}(u) = 3, deg_{H'}(u) \geq 2$ by claim 3. Consider the underlying grid of which G is a subgraph. Pick the edge $(u, v) \in E_{H'}$ which is either not axis-parallel or is axis-parallel and v is at distance at least 2 in the grid. Since u has degree 3 in G', there exists a neighbor w_1 of u in G' such that any path from w_1 to v in G' has length at least 3.

If w_1 has degree 2 in G', then we set w_2 to be the neighbor of w_1 other than u. It is then straightforward to check that u, v, w_1, w_2 satisfy the condition of Lemma 7. Hence, there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

If w_1 is not of degree 2 in G', it must have degree 3 since there are no degree 1 vertices in G'. Let w_2 be the neighbor of w_1 other than u such that $(u, w_2) \notin E'_H$. Such a neighbor exists since u has at most one neighbor in H' other than v and v is not incident to w_1 due to the fact that any path of length from w_1 to v has length at least 3. Let w_3 be the neighbor of w_2 other than w_1. Such a neighbor must exist since there is no vertex of degree 1 in G'. Now, we prove that these vertices satisfy the condition of Lemma 6. By construction $(u, w_1), (w_1, w_2), (w_2, w_3)$ are black eges. Any path from w_1 to v in G' has length at least 3 which implies that $(w_1, v), (w_2, v)$ are not black edges. Also, by construction, (u, v) is a blue edge and (u, w_2) is not a blue edge. Hence, by lemma 6 there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$. ◀

▶ **Lemma 9.** *If there exists a set S of vertices such that no edges leave S in G', but at least one edge leaves S in H' then then there exists a vector $x \in \mathbb{R}^n$ such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.*

Proof. Let x be defined as follows: $x_u = 1$ for $u \in S$ and $x_u = 0$ for $u \notin S$. The $\mathcal{R}(G', x)$ is equal to the number of edges leaving S in G', and similarly for H'. The lemma follows. ◀

▶ **Lemma 10.** *If there exists a cycle of length more than 4 in G' such that all vertices of cycle have degree 2 in G', then there exists $x \in \mathbb{R}^n$ such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.*

Proof. Let C be the set of vertices in the cycle. For any vertex v in C, $deg_{G'}(v) = 2$. By claim 3, $deg_{H'}(v) \geq 1$. If there is no blue edge connecting two vertices of C in G', then there are at least $|C|$ edges going out of C in H' and no edge going out of C in G'. Then, by lemma 9, there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

In the complementary case, suppose there is an edge $(a, b) \in E_{H'}$ such that $a, b \in C$. Let the two paths from a to b on C be $P_1 = a, w_1, \ldots, w_{k_1}, b$ and $P_2 = a, v_1, \ldots, v_{k_2}, b$. Since G' and H' do not share any edge, $\min(k_1, k_2) \geq 1$. And since the cycle has length at least 5, $\max(k_1, k_2) \geq 2$. Let a vector x be defined as follows: $x_a = 0, x_b = 1, x_{w_i} = \frac{i}{k_1+1}, x_{v_i} = \frac{i}{k_2+1}$ and for $u \notin C$, set $x_u = 0$. We have

$$\mathcal{R}(G', x) \;=\; \sum_{(u,v)\in P_1} (x_u - x_v)^2 + \sum_{(u,v)\in P_2} (x_u - x_v)^2$$

$$=\; (k_1 + 1) \cdot \frac{1}{(k_1 + 1)^2} + (k_2 + 1) \cdot \frac{1}{(k_2 + 1)^2} \leq \frac{5}{6}.$$

The last inequality holds because $\min(k_1, k_2) \geq 1, \max(k_1, k_2) \geq 2$.
On the other hand $\mathcal{R}(H', x) \geq (x_a - x_b)^2 = 1$ and the lemma follows. ◀

▶ **Lemma 11.** *If G' contains a set of disjoint cycles of length 4 and H' edges only have endpoints on the same cycle, then there is a cycle $\tilde{H}$ such that $|E_{\tilde{H}} \cap E_G| = |E_H \cap E_G| + 2$.*

Proof. Consider a cycle C of length 4 in G' such that there is no blue edge between a vertex in the cycle and a vertex not in the cycle. Since $deg_{G'}(v) \geq 2$ for all vertices in the cycle, $deg_{H'}(v) \geq 1$ by claim 3. Since G' and H' do not share any edge and the cycle has length 4, we must have $deg_{H'}(v) = 1$ for all $v \in C$. And for vertices not in the length 4 cycles, $deg_{H'}(v) = 0$. Let the edges of H be $F_1 \cup F_2$ where F_1 are the edges shared between G and H and F_2 are the diagonal edges in the disjoint cycles of length 4. Let C be one such cycle in G' with vertices v_1, v_2, v_3, v_4 in this order and $(v_1, v_3) \in F_2, (v_2, v_4) \in F_2$. Let $H_1 = (V, F_1 \cup F_2 \backslash \{(v_1, v_3), (v_2, v_4)\} \cup \{(v_1, v_2), (v_3, v_4)\}), H_2 = (V, F_1 \cup F_2 \backslash \{(v_1, v_3), (v_2, v_4)\} \cup \{(v_1, v_4), (v_2, v_3)\})$. Then, one of the H_1 or H_2 is a cycle of length n. We let $\tilde{H}$ be that cycle. ◀

Finishing the proof: Recall that we have assumed that G does not contain a Hamiltonian cycle and let H is a permutation of C such that $|E_G \cap E_H|$ is maximized. To show that G does **not** dominate H, we need to construct a vector x such that $\mathcal{R}(H, x) > \mathcal{R}(G, x)$.

Starting with G and H, we form two graphs G' and H' as follows: (i) delete from G and H all common edges, (ii) iteratively and greedily delete all vertices such that $deg_G(u) = 1, deg_H(u) = 0$. Then Claims 1 and 2 show that it suffices to find a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

If there is still a vertex with $deg_{G'}(u) = 1$, then $deg_{H'}(u)$ must be at least 1 and hence, by Claim 4, there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$. Also, if there is a vertex u with degree 3 in G', then by lemma 8 there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

If there are no degree 1 or degree 3 vertices G', then G' must be a collection of isolated vertices and cycles. If there is a vertex v such that $deg_{G'}(v) = 0, deg_{H'}(v) \geq 1$, then by setting $S = \{v\}$, lemma 9 implies that there exists a vector x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

So, if none of the above cases occurs, then G' is a collection of disjoint cycles and H' edges are only incident to vertices of the cycles. If there is a cycle of length at least 5, then by lemma 10 there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$. Otherwise, if there is at least one blue edge with end points on two different cycles of length 4, then by setting S to be the vertex set of the cycle of length 4 Lemma 9 implies that there exists x such that $\mathcal{R}(H', x) > \mathcal{R}(G', x)$.

So, either G' and H' are empty or G' consists of a collection of disjoint cycles of length 4 such that blue edges have end points in the same cycle. In the first case, G trivially contains a Hamiltonian cycle since H' is empty. This is a contradiction to the assumption that G does not contain a Hamiltonian cycle. In the second case G Lemma 11 contradicts our assumption about the maximality of $|E_G \cap E_H|$. ◀

Proof. (Theorem 4) The problem of detecting if a cubic subgrid contains a Hamiltonian cycle is NP-complete [17]. Hence Theorem 5 is a direct reduction, and the theorem follows. ◀

3 Spectrally Robust Graph Isomorphism on Trees

This section outlines a proof for Theorem 3. We first introduce necessary notation.

▶ **Definition 12.** The support $\sigma(G, H)$ of G by H is the smallest number γ such that γH dominates G.

▶ **Definition 13.** The condition $\kappa(G, H)$ of a pair of graphs G and H is the smallest number κ such that G and H are κ-similar. We have $\kappa(G, H) = \sigma(G, H)\sigma(H, G)$.

We denote by $d_G(u, v)$ the distance between u and v in G using the shortest path metric. For $S \subseteq V$, we denote by $\delta_G(S)$ the set of of edges crossing the cut $(S, V - S)$ in G.

We now briefly review well-known facts [8]; a more detailed version of this paragraph along with proofs can be found in the appendix of the full version of the paper [12]. Given two trees G and H there is an obvious way to embed the edges of G into H: each edge (u, v) is routed over the unique path between vertices (u, v) in H. The *dilation* of the embedding is defined by: $\mathbf{d} = \max_{(u,v) \in E_G} d_H(u, v)$. The congestion c_e of an edge $e \in E_H$ is the number of G-edges that are routed over e. The *congestion* $\mathbf{c}$ of the embedding is then defined as $\max_{e \in E_H} c_e$. An upper bound of κ on the condition number implies the same upper bound on both c and d. On the other hand, the product $\mathbf{cd}$ is an upper bound on $\sigma(G, H)$, which is at most a quadratic over-estimation of $\sigma(G, H)$.

Our algorithm finds a mapping that controls both the dilation and the congestion of the embeddings from G to H and vice versa, thus obtaining a quadratic approximation to the condition number as a corollary.

▶ Remark. To simplify the presentation and the proof, we assume uniform upper bounds on the congestion and the dilation of both embeddings (G to H and H to G), rather than handling them separately. This formally proves a κ^3-approximation to κ-similarity. A κ-approximation algorithm may be possible with a refined argument that tracks the two embeddings separately.

Formally, our result can be stated as follows:

▶ **Theorem 14.** *Suppose G and H are two trees for which there exists a bijective mapping $\pi : V(G) \to V(H)$ satisfying the following properties:*
- *For all $(u, v) \in E(G)$, $d_H(\pi(u), \pi(v)) \leq \ell$*
- *For all $(u, v) \in E(H)$, $d_G(\pi^{-1}(u), \pi^{-1}(v)) \leq \ell$*
- *For $S \subset V(G)$ such that $|\delta_G(S)| = 1$, $|\delta_H(\pi(S))| \leq k$.*
- *For $S \subset V(H)$ such that $|\delta_H(S)| = 1$, $|\delta_G(\pi^{-1}(S))| \leq k$.*

Then, there exists an algorithm to find such a mapping in time $n^{O(k^2 d)}$ where d is the maximum degree of a vertex in G or H.

Our main result, theorem 3, follows immediately as a corollary from the fact that $\max\{k, l\} \leq \sigma(H, G) \leq kl$ (which is proved in the full version of the paper) and using the fact that $\max\{\sigma(G, H), \sigma(H, G)\} \leq \kappa(G, H) \leq \sigma(G, H)\sigma(H, G)$.

▶ **Corollary 15.** *Given two tree graphs G and H with condition number κ and maximum degree d, there exists an algorithm running in time $n^{O(\kappa^2 d)}$ which finds a mapping certifying that condition number is at most κ^4.*

The algorithm uses dynamic programming; it proceeds by recursively finding mappings for different subtrees and merging them. The challenge is to find partial mappings of subtrees which also map their boundaries in such a way that enables different mappings to

be appropriately merged. Notice that it is not enough to consider just the boundary vertices of the subtrees and their images. Instead, we need to additionally consider the boundary edges of those vertex sets, which correspond to cuts induced on the graph.

Definitions and Lemmas.   To proceed with the proof, we introduce some definitions. We fix k and ℓ to be defined as in Theorem 14. Also, we fix an arbitrary ordering L on the edges of H. Without loss of generality it is convenient to root the trees such that we always map the two roots to each other. Let r_G be the root of tree G and r_H be the root of tree H.

Suppose that u is a vertex in G and T_u^G is the subtree rooted at u in G. If T_u^G is mapped to the set T in H, then its boundary includes the vertex u, the edge incident to u, the boundary vertices of T, and the cuts induced by edges going out of T. Hence, in addition to considering the mapping of boundary vertices, we need to consider the mapping of sets T' such that $\delta_H(T') = \{e\}$ where $e \in \delta_H(T)$. This notion is formalized in the following two definitions.

▶ **Definition 16.** Let Γ be the set of tuples $(u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x)$ satisfying the following properties:
- $u, u_1, \ldots, u_x \in V(G), v \in V(H), T \subset V(H), S_1, \ldots, S_x \subset V(G)$;
- $r_G \notin S_1, \ldots, S_x, r_H \notin T$;
- $u, u_1, \ldots, u_x \neq r_G, v \neq r_H$;
- $|\delta_H(T)| = x \leq k$ and $\forall j \in [1, x], |\delta_G(S_j)| \leq k$.

For $\alpha \in \Gamma$, we use the indicator variable z_α to denote if there is a mapping π which realizes α and preserves the distances and cuts for edges in T_u^G and T. A permutation π realizing α is formally defined below. Intuitively, this mapping maps the subtree rooted at u in G to the set T in H, vertex u to vertex v. It also maps $u_1, \ldots, u_x$ to the vertex boundary of the set T, and maps sets $S_1, \ldots, S_x$ to the cuts induced by the boundary edges of T. The formal definition of z_α is as follows:

▶ **Definition 17.** For $\alpha = (u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x) \in \Gamma$, let $\delta_H(T) = \{e_1, \ldots, e_x\}$ be such that for $i < j$, e_i is ordered before e_j in ordering L. Let $v_j = e_j \cap T$, T_u^G be the vertex set in the subtree rooted at u in G and for $e \in E(G)$, let T_e^G be the vertex set in the subtree under edge e. Formally speaking $T_e^G \subset V(G)$ such that $\delta_G(T_e^G) = \{e\}$ and $r_G \notin T_e^G$ (T_e^H is similarly defined). We define $z_\alpha = 1$ if there exists a mapping $\pi : V(G) \to V(H)$ such that:
1. $\pi(T_u^G) = T, \pi(u) = v, \forall j \in [1, x], \pi(u_j) = v_j, \pi(S_j) = T_{e_j}^H, \pi(r_G) = r_H$.
2. $\forall (u, v) \in E[G[T_u^G]], d_H(\pi(u), \pi(v)) \leq \ell$
3. $\forall (u, v) \in E[H[T]], d_G(\pi^{-1}(u), \pi^{-1}(v)) \leq \ell$
4. $\forall e \in E[G[T_u^G]], |\delta_H(\pi(T_e^G)| \leq k$.
5. $\forall e \in E[H[T]], |\delta_G(\pi^{-1}(T_e^H)| \leq k$.
We refer to such a mapping π as a certificate of $z_\alpha = 1$. Moreover, we define $z_{\alpha,\pi} = 1$ if π is a certificate of $z_\alpha = 1$ and 0 otherwise.

▶ **Claim 5.** *There exists a poly(n) time algorithm which given $\alpha \in \Gamma, \pi : V(G) \to V(H)$, outputs the value of $z_{\alpha,\pi}$.*

Our goal is to design an algorithm which computes z_α for every $\alpha \in \Gamma$. However, for our algorithm to run in polynomial time, we need Γ to not be exponentially large.

▶ **Lemma 18.** $|\Gamma| \leq n^{O(k^2)}$.

Proof. Let $\alpha = (u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x)$. We prove the lemma by bounding the number of choices for each parameter.

- The number of choices of u is upper bounded by n.
- Since $|\delta_H(T)| = x$, the number of choices $\delta_H(T)$ is upper bounded by $\binom{m}{x}$ where m is the number of edges. By substituting $m = n - 1$, we get that the number of different $\delta_H(T)$, i.e. the number of different T's, is upper bounded by $\binom{n-1}{x}$.
- The number of different v and u_j is at most n for each $j \in [1, x]$.
- Similarly to the argument for T, the number of different S_j with $|\delta_H(S_j)| \le k$ is at most $\sum_{t=1}^{k} \binom{n-1}{t}$.

For $x \in [1, k]$, the number of different tuples α in Γ with $|\delta_H(T)| = x$ is at most:

$$n \cdot \binom{n-1}{x} n \cdot n^x \cdot \left[\sum_{t=1}^{k} \binom{n-1}{t} \right]^x = n^{O(k \cdot x)}.$$

Since $x \le k$, this gives us an upper bound of $n^{O(k^2)}$ on $|\Gamma|$.   ◀

Suppose π is the optimal mapping from G to H which yields a mapping with cut distortion k and distance distortion ℓ and also certifies $z_\alpha = 1$ for some α. Our recursive algorithm does not necessarily obtain the same certificate as π for $z_\alpha = 1$. So, before we show how to compute z_α, we examine certain properties of z_α. In particular, we start by proving that if both π and γ certify $z_\alpha = 1$ so that $z_{\alpha,\pi} = z_{\alpha,\gamma} = 1$, then they not only match on the boundary vertices but also on the cuts induced by boundary edges.

▶ **Lemma 19.** *For $\alpha = (u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x)$, let π and γ be two mappings such that $z_{\alpha,\pi} = z_{\alpha,\gamma} = 1$. Then:*
1. $\pi(u) = \gamma(u)$.
2. $\pi(T_u^G) = \gamma(T_u^G)$.
3. *For every boundary vertex w of T (in T with an incident edge in $\delta_H(T)$), $\pi^{-1}(w) = \gamma^{-1}(w)$. Equivalently, $\pi(u_j) = \gamma(u_j)$ for $j \in [1, x]$.*
4. *For every edge $e \in \delta_H(T)$, $\pi^{-1}(T_e^H) = \gamma^{-1}(T_e^H)$.*
5. *For every connected component C in $H \setminus \delta_H(T)$, $\pi^{-1}(C) = \gamma^{-1}(C)$.*

Proof. Items 1-4. follow directly from the definition of $z_{\alpha,\pi}$. Consider a connected component C in $H \setminus \delta_H(T)$. Let $\delta_H(C) = \{e_{i_1}, \ldots, e_{i_t}\}$. Without loss of generality, assume that e_{i_1} is the edge closest to the root r_H. Then:

$$\gamma^{-1}(C) = \gamma^{-1}(T_{e_{i_1}}) \setminus \cup_{j=2}^{t} \gamma^{-1}(T_{e_{i_j}}).$$

Item 3 implies that $\gamma^{-1}(T_{e_{i_j}}) = \pi^{-1}(T_{e_{i_j}})$ for $j \in [1, t]$ thus proving $\pi^{-1}(C) = \gamma^{-1}(C)$.   ◀

The next lemma is somewhat like a converse of the previous lemma. It shows that if we have a mapping π such that $z_{\alpha,\pi} = 1$ and another mapping γ such that γ matches with π on the subtree and the boundary vertices and edges, then $z_{\alpha,\gamma} = 1$ as well.

▶ **Lemma 20.** *Let $\alpha = (u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x)$ and $\pi : V(G) \to V(H)$ be such that $z_{\alpha,\pi} = 1$. Let $\gamma : V(G) \to V(H)$ be such that*
1. $\gamma(w) = \pi(w)$ for $w \in T_u^G$
2. $\gamma(u_j) = \pi(u_j)$ for $j \in [1, x]$
3. $\gamma(S_j) = \pi(S_j)$ for $j \in [1, x]$ (π and γ may not be identical on every element of S_j)
Then, $z_{\alpha,\gamma} = 1$.

Proof. Follows immediately from definition 17.   ◀

Next we show how to change the optimal mapping such that it agrees with the mapping found by our algorithm on the subtree and is still optimal. Following lemma formalizes this statement:

▶ **Lemma 21.** *Let π be a mapping such that $z_{\alpha,\pi} = z_{\alpha_1,\pi} = 1$ where*

$$\alpha = (a, T, b, u_1, \ldots, u_x, S_1, \ldots, S_x) \in \Gamma \text{ and } \alpha_1 = (a^1, T^1, b^1, u_1^1, \ldots, u_{x_1}^1, S_1^1, \ldots, S_{x_1}^1) \in \Gamma$$

such that a_1 is a child of a in G. Suppose, γ_1 is another mapping such that $z_{\alpha_1,\gamma_1} = 1$. Let ζ be defined as follows: $\zeta(u) = \gamma_1(u)$ for $u \in T_{a^1}^G$ and $\zeta(u) = \pi(u)$ otherwise. Then, $z_{\alpha,\zeta} = 1$.

Proof. Items 1-4 of lemma 17 can be easily verified for $z_{\alpha,\zeta}$. Here, we prove that the following property holds: $\forall e \in E[H[T]], |\delta_G(\zeta^{-1}(T_e^H))| \leq k$.

There are three possible cases, that we consider separately.

1. $e = (b, b_1)$, $e \in \delta_H(T^1)$.
   By definition, $\zeta^{-1}(T_e^H) = \gamma_1^{-1}(T_e^H)$. Since, $z_{\alpha_1,\gamma_1} = 1$, we get $|\zeta^{-1}(T_e^H)| = |\gamma_1^{-1}(T_e^H)| \leq k$.

2. $e \in E[H[T \setminus T^1]]$.
   By definition, $\zeta^{-1}(T_e^H) = \pi^{-1}(T_e^H)$. Since, $z_{\alpha,\pi} = 1$,, we get $|\zeta^{-1}(T_e^H)| = |\pi^{-1}(T_e^H)| \leq k$.

3. $e \in E[H[T^1]]$.
   By definition, $\zeta^{-1}(T_e^H) = \gamma_1^{-1}(T_e^H)$ and since, $z_{\alpha_1,\gamma_1} = 1$, we get $|\zeta^{-1}(T_e^H)| \leq k$.   ◀

The above lemmas show that even if we find mappings for subtrees which are different from the optimal mappings, they can still be merged with the optimal mappings. Hence, we may just find any of the mappings for each α and then recursively combine mappings. The following lemma states the result and shows how to make it constructive:

▶ **Lemma 22.** *Let $a \in V(G)$ be a vertex in G with children $a^1, \ldots, a^t$. Let*

$$\alpha = (a, T, b, u_1, \ldots, u_x, S_1, \ldots, S_x) \text{ and}$$
$$\alpha_1 = (a^1, T^1, b^1, u_1^1, \ldots, u_{x_1}^1, S_1^1, \ldots, S_{x_1}^1), \ldots, \alpha_t = (a^t, T^t, b^t, u_1^t, \ldots, u_{x_t}^t, S_1^t, \ldots, S_{x_t}^t) \in \Gamma.$$

Let $\pi : V(G) \to V(H)$ be a mapping such that $z_{\alpha,\pi} = z_{\alpha_1,\pi} = \cdots = z_{\alpha_j,\pi} = 1$ for all $j \in [1, t]$. If for each $j \in [1, t]$ there exists a mapping $\gamma_j : V(G) \to V(H)$ with $z_{\alpha_j,\gamma_j} = 1$, then there exists $\pi' : V(G) \to V(H)$ such that $z_{\alpha,\pi'} = 1$ and $\pi'(w) = \gamma_j(w)$ for $w \in T_{a_j}^G$ where $j \in [1, t]$. Moreover, given $\{\gamma_j \mid j \in [1, t]\}$, such π' can be found in time poly(n).

Proof. Let $\zeta : V(G) \to V(H)$ such that $\zeta(w) = \gamma_j(w)$ for $w \in T_{a_j}^G$ where $j \in [1, t]$ and $\zeta(w) = \pi(w)$ otherwise. By lemma 21, $z_{\alpha,\zeta} = 1$.

Construction of π': Let $\pi'(w) = \gamma_j(w)$ for $w \in T_{a_j}^G, j \in [1, t]$ and $\pi'(a) = b$. For $w \notin T_a^G$, define π' such that $\pi'(S_j) = T_j$ for $j \in [1, x]$. Setting $\pi = \zeta, \gamma = \pi'$ in lemma 20, we get $z_{\alpha,\pi'} = 1$. Easy to see that π' is constructed in polynomial time.   ◀

Lemma 22 suggests that we can recursively compute z_α. Namely, we can show the following:

▶ **Lemma 23.** *There exists an algorithm with running time poly$(n, |\Gamma|^d)$ which calculates z_α for each $\alpha \in \Gamma$. Additionally if $z_\alpha = 1$, it also computes π_α such that $z_{\alpha,\pi_\alpha} = 1$.*

Proof. Consider $\alpha = (a, T, b, u_1, \ldots, u_x, S_1, \ldots, S_x) \in \Gamma$ with $z_\alpha = 1$ and $\pi : V(G) \to V(H)$ be the mapping such that $z_{\alpha,\pi} = 1$. Let the children of a be $a^1, \ldots, a^t$.

▶ **Claim 6.** *For $j \in [1, t], \exists \alpha_j = (a^j, T^j, b^j, u_j^1, \ldots, u_{x_j}^1, S_1^1, \ldots, S_{x_j}^1) \in \Gamma$ such that $z_{\alpha_j,\pi} = 1$.*

To construct a mapping π' such that $z_{\alpha,\pi'} = 1$, we guess $\alpha_1, \ldots, \alpha_t$ and use lemma 22 to construct such a mapping. It requires mapping γ_j such that $z_{\alpha_j,\gamma_j} = 1$ which can be assumed to be constructed recursively. The number of choices of $\alpha_1, \ldots, \alpha_t$ is upper bounded by $|\Gamma|^t$ which is upper bounded by $|\Gamma|^d$, as the degree of any vertex is at most d. For any such choice, algorithm in lemma 22 runs in time $poly(n)$. Hence, computing z_α takes $|\Gamma|^d poly(n)$ time for each α and $|\Gamma|^{d+1} poly(n) = poly(n, |\Gamma|^d)$ time for all $\alpha \in \Gamma$.

If $z_\alpha = 0$, then for any of the mappings π' considered above has $z_{\alpha,\pi'} = 0$. This can be checked in $poly(n)$ time for each π' (proposition 5). ◀

Proof. (of theorem 14) Let π be a mapping $\pi : V(G) \to V(H)$ such that:
(a) For all $(u, v) \in E(G)$, $d_H(\pi(u), \pi(v)) \leq \ell$
(b) For all $(u, v) \in E(H)$, $d_G(\pi^{-1}(u), \pi^{-1}(v)) \leq \ell$
(c) For $S \subset V(G)$ s.t. $|\delta_G(S)| = 1$, $|\delta_H(\pi(S))| \leq k$.
(d) For $S \subset V(H)$ s.t. $|\delta_H(S)| = 1$, $|\delta_G(\pi^{-1}(S))| \leq k$.

First, we start by guessing the roots of G and H and define Γ. Then, using lemma 23, we can calculate z_α for $\alpha \in \Gamma$. It does not give us a mapping π' satisfying the conditions above since for $\alpha = (u, T, v, u_1, \ldots, u_x, S_1, \ldots, S_x) \in \Gamma$, we have $u \neq r_G$. However, a proof almost identical to that of lemma 23 works here as well. Assume r_G has children $a_1, \ldots, a_t$. Then there exists $\alpha_j = (a^j, T^j, b^j, u_j^1, \ldots, u_{x_j}^1, S_1^1, \ldots, S_{x_j}^1) \in \Gamma$ such that $z_{\alpha_j,\pi} = 1$. Then, similarly to the proof of lemma 23 we can guess $\alpha_j, j \in [1, t]$ and compute π' in time $poly(n) \cdot |\Gamma|^d$, which satisfies the conditions described above. ◀

4   Final Remarks

From an algebraic standpoint, the problems we considered in this work have natural generalizations to pairs of positive definite matrices (A, B), and the corresponding eigenvalue problem $Ax = \lambda P^T B P x$. SGD generalizes to minimizing the maximum eigenvalue, and SGRI generalizes to finding the permutation P that minimizes the condition number $\kappa(A, B)$. But the problem appears to be much harder in some sense: one can construct 'pathological' examples of A and B with just two distinct eigenspaces that are nearly identical, but different enough to cause unbounded condition numbers due to the eigenvalue gap. This makes implausible the existence of non-trivial subexponential time algorithms for the general case.

On the other hand, besides their potential for applications, Laplacians seem to offer an interesting mathematical ground with a wealth of open problems. In this paper we presented the first algorithmic result, for unweighted trees. The algorithm is admittedly complicated, but it can at least be viewed as an indication of algorithmic potential, as we are not aware of any fact that would preclude a similar approximation for general graphs. To make such algorithmic progress, we would likely have to give up on the combinatorial interpretations of the condition number, and use deeper spectral properties of Laplacians.

References

1   László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 684–697, 2016.

2   Joshua Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: Theory and algorithms. *Commun. ACM*, 56(8):87–94, 2013. `doi:10.1145/2492007.2492029`.

3 Adolfo Piperno Brendan D. Mckay. Faster isomorphism testing of strongly regular graphs. *Journal of Symbolic Computation archive*, 60:94–112, 2014.

4 P. G. Doyle and J. L. Snell. Random Walks and Electric Networks. *ArXiv Mathematics e-prints*, 2000. `arXiv:math/0001057`.

5 Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. Fifty years of graph matching, network alignment and network comparison. *Inf. Sci.*, 346(C):180–197, 2016. `doi:10.1016/j.ins.2016.01.074`.

6 Soheil Feizi, Gerald Quon, Mariana Recamonde Mendoza, Muriel Médard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of networks. *CoRR*, abs/1602.04181, 2016. `arXiv:1602.04181`.

7 S. Fortin. The graph isomorphism problem. *Technical Report 96-20*, 1996.

8 Stephen Guattery and Gary L. Miller. Graph embeddings and laplacian eigenvalues. *SIAM J. Matrix Analysis Applications*, 21(3):703–723, 2000.

9 Israel Hanukoglu. Electron transfer proteins of cytochrome p450 systems. In E. Edward Bittar, editor, *Physiological Functions of Cytochrome P450 in Relation to Structure and Regulation*, volume 14 of *Advances in Molecular and Cell Biology*, pages 29–56. Elsevier, 1996. `doi:10.1016/S1569-2558(08)60339-2`.

10 Claire Kenyon, Yuval Rabani, and Alistair Sinclair. Low distortion maps between point sets. In *SIAM J. Comput*, volume 39(4), page 1617–1636, 2004.

11 J. Kobler, U. Schoning, and J. Toran. The graph isomorphism problem: Its structural complexity. *Progress in Theoretical Computer Science*, 1993.

12 Alexandra Kolla, Ioannis Koutis, Vivek Madan, and Ali Kemal Sinop. Spectrally robust graph isomorphism. *CoRR*, abs/1805.00181, 2018. URL: `https://arxiv.org/abs/1805.00181`.

13 Ioannis Koutis. *Combinatorial and algebraic tools for optimal multilevel algorithms*. PhD thesis, Carnegie Mellon University, Pittsburgh, May 2007. CMU CS Tech Report CMU-CS-07-131.

14 Ioannis Koutis, Gary L. Miller, and Richard Peng. A fast solver for a class of linear systems. *Commun. ACM*, 55(10):99–107, 2012. `doi:10.1145/2347736.2347759`.

15 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.*, 25(1):42–65, 1982.

16 Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, lasserre gaps, and asymmetry of random graphs. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1659–1677, 2014.

17 Christos H Papadimitriou and Umesh V Vazirani. On two geometric problems related to the travelling salesman problem. *Journal of Algorithms*, 5(2):231–246, 1984.

18 Robert Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012. `doi:10.1093/bioinformatics/bts592`.

19 Daniel A. Spielman. Faster isomorphism testing of strongly regular graphs. *Proc. 28th ACM STOC*, 215:576–584, 1995.

20 Charalampos E. Tsourakakis. Toward quantifying vertex similarity in networks. *Internet Mathematics*, 10(3-4):263–286, 2014. `doi:10.1080/15427951.2013.836581`.

21 Regina I. Tyshkevich Viktor N. Zemlyachenko, Nikolai M. Korneenko. Graph isomorphism problem. *Zapiski Nauchnykh Seminarov LOMI*, 215:83–158, 1982.