On the Complexity of the Eigenvalue Deletion Problem

Neeldhara Misra

Indian Institute of Technology, Gandhinagar, India

Harshil Mittal

Indian Institute of Technology, Gandhinagar, India

Saket Saurabh

Institute of Mathematical Sciences, Chennai, India

University of Bergen, Norway

Dhara Thakkar

Indian Institute of Technology, Gandhinagar, India

Abstract

For any fixed positive integer r and a given budget k, the r-EIGENVALUE VERTEX DELETION (r-EVD) problem asks if a graph G admits a subset S of at most k vertices such that the adjacency matrix of $G \setminus S$ has at most r distinct eigenvalues. The edge deletion, edge addition, and edge editing variants are defined analogously. For $r = 1$, r-EVD is equivalent to the Vertex Cover problem. For $r = 2$, it turns out that r-EVD amounts to removing a subset S of at most k vertices so that $G \setminus S$ is a cluster graph where all connected components have the same size.

We show that r-EVD is NP-complete even on bipartite graphs with maximum degree four for every fixed $r > 2$, and FPT when parameterized by the solution size and the maximum degree of the graph.

We also establish several results for the special case when $r = 2$. For the vertex deletion variant, we show that 2-EVD is NP-complete even on triangle-free and $3d$-regular graphs for any $d \geqslant 2$, and also NP-complete on d-regular graphs for any $d \geqslant 8$. The edge deletion, addition, and editing variants are all NP-complete for $r = 2$. The edge deletion problem admits a polynomial time algorithm if the input is a cluster graph, while – in contrast – the edge addition variant is hard even when the input is a cluster graph. We show that the edge addition variant has a quadratic kernel. The edge deletion and vertex deletion variants admit a single-exponential FPT algorithm when parameterized by the solution size alone.

Our main contribution is to develop the complexity landscape for the problem of modifying a graph with the aim of reducing the number of distinct eigenvalues in the spectrum of its adjacency matrix. It turns out that this captures, apart from Vertex Cover, also a natural variation of the problem of modifying to a cluster graph as a special case, which we believe may be of independent interest.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Graph Modification, Rank Reduction, Eigenvalues

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.53

Related Version *Full Version*: `https://arxiv.org/abs/2310.00600`

Funding *Neeldhara Misra*: Supported by DST-SERB and IIT Gandhinagar.
Harshil Mittal: Supported by IIT Gandhinagar.
Saket Saurabh: Supported by ERC, the University of Bergen, and IMSc.
Dhara Thakkar: Supported by CSIR-UGC NET JRF Fellowship.

Acknowledgements We thank Daniel Lokshtanov for helpful discussions.

1 Introduction

Graph modification problems are a fundamental class of optimization problems where we have a class of graphs $\mathcal{F}$ that satisfy some property of interest P, the input is a graph G, and we are interested in a smallest subset of vertices $S \subseteq V(G)$ such that $G \setminus S \in \mathcal{F}$. This is a rather general framework that captures several classical optimization problems as special cases, for instance:

- when $\mathcal{F}$ is the collection of edgeless graphs, then the problem is Vertex Cover;
- when $\mathcal{F}$ is the collection of acyclic graphs, then the problem is Feedback Vertex Set;
- when $\mathcal{F}$ is the collection of bipartite graphs, then the problem is Odd Cycle Traversal;

and so on. It has also been of interest to study modifications other than vertex deletion: the most common alternate modifications considered include edge deletion, edge addition, and edge editing (adding and removing edges). The optimization problems for these operations may be posed analogously.

Meesum, Misra, and Saurabh [9] pose the question of modifying a graph with the goal of reducing the rank of the associated adjacency matrix, which is to say that $\mathcal{F}_{\leqslant r}$ is the class of graphs whose adjacency matrices have rank at most r. We use A_G to denote the adjacency matrix of a graph G, and we use the phrase "spectrum of G" to refer to the (multi-)set of eigenvalues of A_G. Previous works focus separately on the settings of undirected [9] and directed [10] graphs.

In the setting of simple undirected graphs, Meesum, Misra, and Saurabh [9] introduce and study the r-Rank Vertex Deletion, r-Rank Edge Deletion, and r-Rank Editing problems. These problems generalize the classical Vertex Cover problem. They show that all the three problems are NP-complete, and are fixed parameter tractable (FPT) in the standard parameter: in particular, they demonstrate an algorithm with running time $2^{\mathcal{O}(k \log r)} n^{\mathcal{O}(1)}$ for r-Rank Vertex Deletion, and an algorithm for r-Rank Edge Deletion and r-Rank Editing running in time $2^{\mathcal{O}(f(r)\sqrt{k} \log k)} n^{\mathcal{O}(1)}$, where k is the size of the solution sought. The authors also leave the following question open:

"[...] what is complexity of the problem of reducing the number of distinct eigenvalues of a graph by deleting a few vertices or editing a few edges?"

In this paper, we address this question at length, developing an initial picture of the complexity landscape for what we call the r-Eigenvalue Vertex Deletion (r-EVD), r-Eigenvalue Edge Deletion (r-EED), r-Eigenvalue Edge Addition (r-EEA), and r-Eigenvalue Edge Editing (r-EEE) problems. All these problems are defined for an arbitrary but fixed positive integer r.

The problem definitions are the following, where we are given an undirected graph G and a positive integer k as input in all cases:

- **r-EVD.** Is there a set $S \subseteq V(G)$ of size $\leqslant k$ such that the number of distinct eigenvalues of $A_{G\setminus S}$ is at most r?
- **r-EEE.** Is there a set $F \subseteq \binom{V(G)}{2}$ of size $\leqslant k$ such that the number of distinct eigenvalues of A_H is at most r, where $H := (V(G), E(G) \Delta F)$?
- **r-EEA.** Is there a set $F \subseteq \binom{V(G)}{2} \setminus E(G)$ of size $\leqslant k$ such that the number of distinct eigenvalues of A_H is at most r, where $H := (V(G), E(G) \cup F)$?
- **r-EED.** Is there a set $F \subseteq E(G)$ of size $\leqslant k$ such that the number of distinct eigenvalues of A_H is at most r, where $H := (V(G), E(G) \setminus F)$?

Note that if we have a solution S for the r-Rank Vertex Deletion problem, then S is also a solution for the $(r+1)$-Eigenvalue Vertex Deletion problem; and analogous statements hold for the other modification problems. This is because the r-Rank Vertex Deletion problem can be equivalently stated as follows: given a graph G and a positive

Table 1 A summary of our results. The result marked † holds for all d except for $d = 1, 2, 3, 4, 5, 7$. Some polynomial cases are omitted from this summary.

	$r = 2$	Fixed $r \geqslant 3$
Vertex Deletion	NP-complete for d-regular graphs † (Theorem 5)	NP-complete even on bipartite graphs (Theorem 8)
	FPT in k (Theorem 6)	FPT in k and $\Delta(G)$ (Theorem 9)
	Polynomial time on forests (Proposition 7)	
Edge Addition	NP-complete even on cluster graphs (Theorem 10)	NP-complete (Theorem 12)
	Quadratic kernel in k (Theorem 11)	
Edge Deletion	NP-complete (Theorem 16)	NP-complete (Theorem 15)
	FPT in k (Theorem 13)	
	Polynomial time on triangle-free graphs (Proposition 14)	
Edge Editing	NP-complete (Theorem 16)	OPEN

integer k, find a smallest subset of vertices $S \subseteq V(G)$ such that $G \setminus S$ has at most r non-zero eigenvalues. However, the converse is not true (since, in general, bounding the number of distinct eigenvalues is not sufficient to bound the rank), making the eigenvalue deletion problems distinct from their rank deletion counterpart.

Our Contributions. We summarize our contributions below, and also in Table 1. We first focus on the special case when $r = 2$. It is known that the adjacency matrix A_G of a graph G has at most two distinct eigenvalues if and only if G is a disjoint union of equal-sized cliques (Lemma 1). Based on this, note that the 2-EIGENVALUE VERTEX DELETION problem is equivalent to finding a subset $S \subseteq V(G)$ of vertices such that $G \setminus S$ is a disjoint union of cliques of size ℓ for some $1 \leqslant \ell \leqslant |V(G)|$. Note that this is closely related to the CLUSTER VERTEX DELETION problem, which is a well-studied question that involves removing a smallest subset of vertices to obtain a cluster graph. However, to the best of our knowledge, the variant where we demand that the clusters have the same size has not been studied. Our results about the "uniform" version of CLUSTER VERTEX DELETION may therefore be of independent interest.

Our main contributions in the context of vertex deletion are the following results:

- We show that 2-EVD is NP-complete on d-regular graphs for all d except for $d = 1, 2, 3, 4, 5, 7$ (Theorem 5).
- We also give a single-exponential FPT algorithm in the standard parameter (Theorem 6), and show that the problem can be solved in polynomial time on forests and d-regular graphs for $d \leqslant 2$ (Proposition 7).
- Further, for any fixed $r \geqslant 3$, we show that r-EVD is NP-complete on bipartite graphs (Theorem 8) and is FPT in the standard parameter combined with the maximum degree of the graph (Theorem 9).

We now describe our findings for the edge modification variants.

- We show that 2-EEA is already NP-complete when the input is either a cluster graph, a forest, or a collection of cycles (Theorem 10).
- We demonstrate that the problem has a quadratic kernel in the standard parameter (Theorem 11).
- We show that r-EEA is NP-complete for any fixed $r \geqslant 3$ (Theorem 12).
- For the edge deletion variant, we show that r-EED is NP-complete for any fixed $r \geqslant 2$ (Theorems 15 and 16).
- For 2-EED, we have a single-exponential FPT algorithm (Theorem 13) in the standard parameter and a polynomial time algorithm on triangle-free graphs (Proposition 14).
- Finally, for the edge editing variant, we show that 2-Eigenvalue Edge Editing is NP-complete (Theorem 16).

Related Work. As we noted previously, the special case when $r = 2$ is closely related to the problem of modifying to a cluster graph, in which we are allowed to modify the graph such that the resulting graph is cluster i.e., it is disjoint union of cliques. Depending on the modifications allowed, these problems are variously refered to as Cluster Vertex Deletion, Cluster Edge Deletion, Cluster Edge Addition and Cluster Edge Editing. Further, Shamir, Sharan, and Tsur [12] have studied a variant of cluster vertex deletion where they additionally demand that the cluster graph obtained after the modification has at most p components.

Problems related to modifying to a cluster graph are very well-studied because they model the clustering problem in various ways. In a clustering problem we are given various data points with some notion of distance between these points, and it is of interest to group these points into "clusters", where each cluster consists of points that are mutually close with respect to the given distance metric. These scenarios can often be modeled with graphs, and in fact graph structure can often be used to model additional constraints of interest. Given the fundamental importance of clustering, it is no surprise that modifying to cluster graphs has attracted substantial interest in the literature of graph algorithms. We refer the reader to [11] for an overview of results related to cluster modification problems.

Another related problem is the problem of deleting to a graph where the connected components have small diameter. This is known as the s-Club Cluster Vertex Deletion problem [3]. Here, we are given a graph G and two integers $s \geqslant 2$ and $k \geqslant 1$; and the question is if it is possible to remove at most k vertices from G such that each connected component of the resulting graph has diameter at most s. Note that this naturally generalizes the problem of modifying to cluster graphs: indeed, the problem is equivalent to Cluster Vertex Deletion for $s = 1$. The edge modification variants have also been considered and are well-studied.

We note that a solution to the r-Eigenvalue Vertex Deletion problem will also be a valid solution to the $(r-1)$-Club Cluster Vertex Deletion due to Lemma 2, which states that graphs of diameter d have at least $(d+1)$ distinct eigenvalues. This is analogously true for the other modification problems as well. On the other hand, it is easy to see that the converse is not necessarily true.

Remarks. Due to lack of space, we describe most proofs informally and defer a detailed exposition to a full version of the paper. Such results are marked with a $(\star)$. The full version also has a list of problem definitions and extended technical preliminaries. Throughout, we use the $\mathcal{O}^\star(\cdot)$ notation to suppress polynomial factors.

Sections 3,4,5, and 6 focus respectively on the problems of r-EVD, r-EEA, r-EED, and r-EEE.

2 Preliminaries

Let $G = (V, E)$ be a simple undirected graph, where, $V(G)$ is the vertex set of G and $E(G)$ is the edge set of G. We typically use n and m to denote $|V(G)|$ and $|E(G)|$, respectively. The adjacency matrix $A_G = a_{ij}$ of a graph G is an $n \times n$ matrix with $a_{ij} \in \{0, 1\}$ the entry $(i.j) = 1$ if the pair (i, j) is an edge in G. We note that the spectrum of A_G can be computed in polynomial time. A *principal submatrix* of a square matrix A is a matrix obtained by removing an equal number of rows and columns from A such that the indices of the removed rows match with the indices of the removed columns.

The following known results will be relevant to our discussions:

▶ **Lemma 1** ([8, 5]). *Let G be a graph. Then, its adjacency matrix A_G has at most two distinct eigenvalues if and only if G is a disjoint union of equal-sized cliques.*

▶ **Lemma 2** ([2], Proposition 1.3.3). *Let G be a connected graph with diameter d. Then, its adjacency matrix A_G has at least $d + 1$ distinct eigenvalues.*

▶ **Lemma 3** (Cauchy interlacing; [2], Corollary 2.5.2). *Let A be a symmetric matrix of size $n \times n$. Let B be a principal submatrix of A of size $(n - 1) \times (n - 1)$. Then, the eigenvalues of B interlace the eigenvalues of A. That is,*

$$\mu_1 \geqslant \sigma_1 \geqslant \mu_2 \geqslant \sigma_2 \geqslant \mu_3 \geqslant \dots\dots\dots\dots \geqslant \mu_{n-2} \geqslant \sigma_{n-2} \geqslant \mu_{n-1} \geqslant \sigma_{n-1} \geqslant \mu_n$$

where, $\mu_1 \geqslant \mu_2 \geqslant \dots\dots \geqslant \mu_n$ denote the n eigenvalues of A, and $\sigma_1 \geqslant \sigma_2 \geqslant \dots\dots \geqslant \sigma_{n-1}$ denote the $n - 1$ eigenvalues of B.

▶ **Lemma 4** ([2], Chapter 3, Exercise 1). *Let G be a graph with smallest eigenvalue -1. Then, G is a disjoint union of cliques.*

Some examples of graph classes whose spectrum is well-known include complete graphs, paths and cycles ([2], Chapter 1). A complete graph on n vertices has eigenvalues -1 and $n - 1$ (with multiplicities $n - 1$ and 1 respectively). A path on n vertices has eigenvalues $2\cos\left(\frac{\pi j}{n+1}\right)\big|_{1\leqslant j\leqslant n}$. A cycle on n vertices has eigenvalues $2\cos\left(\frac{2\pi j}{n}\right)\big|_{0\leqslant j\leqslant n-1}$.

3 Reducing eigenvalues by deleting vertices

In this section, we show that the r-EVD problem is NP-complete for $r \geqslant 1$. Recall that for $r = 1$, r-EVD is equivalent to VERTEX COVER. For $r = 2$, we show that the problem is NP-complete on general graphs, admits a single-exponential FPT algorithm in the standard parameter, and is polynomial-time solvable on trees. For any fixed $r \geqslant 3$, we show that the problem is NP-complete on bipartite graphs and is FPT in the standard parameter combined with the maximum degree of the graph.

3.1 Deleting to Two Distinct Eigenvalues

Note that by Lemma 1, 2-EVD is equivalent to UNIFORM CLUSTER VERTEX DELETION, a problem where the input is a graph G and a positive integer k and the question is if there is a subset $S \subseteq V(G)$ of vertices such that $G \setminus S$ is a disjoint union of ℓ-sized cliques for some $1 \leqslant \ell \leqslant |V(G)|$. Note that ℓ is not a part of the input. We begin by showing that the problem is hard even when restricted to d-regular graphs for any d other than $1, 2, 3, 4, 5, 7$.

▶ **Theorem 5** ($\star$). *2-EIGENVALUE VERTEX DELETION is NP-complete even on triangle-free and $3d$-regular graphs for any $d \geqslant 2$, and NP-complete on d-regular graphs for any $d \geqslant 8$.*

To show this result we use two reductions: one from the INDEPENDENT SET problem on cubic triangle-free graphs and the other from INDEPENDENT SET on planar cubic triangle-free graphs.

In the first construction, we replace every vertex v with vertices $v^{(1)}$ and $v^{(2)}$, and extended the edges as follows: an edge (u, v) maps to the edges $(u^{(1)}, v^{(1)})$, $(u^{(1)}, v^{(2)})$, $(u^{(2)}, v^{(1)})$, and $(u^{(2)}, v^{(2)})$. Note that this construction preserves triangle-freeness and transforms a cubic graph to a six-regular graph. For demonstrating hardness on $3d$ regular graphs for $d \geqslant 2$, we make d copies of the vertices instead of two copies.

For the second construction, we make six copies of the graph and for every vertex, we induce a clique on all its copies. This construction turns a cubic graph into a 8-regular graph. For demonstrating hardness on d regular graphs for $d \geqslant 8$, we make $(d-2)$ copies of the vertices instead of six.

Next, we note that 2-EVD admits a branch-and-bound-based FPT algorithm that is similar in spirit to the naive branching algorithm for CLUSTER VERTEX DELETION. As long as our instance has an induced path on three vertices $\{u, v, w\}$, we recursively solve the instances $(G \setminus \{u\}, k-1)$, $(G \setminus \{v\}, k-1)$ and $(G \setminus \{w\}, k-1)$. Note that this branching algorithm enumerates all minimal subsets S of size at most k such that $G \setminus S$ is a disjoint union of cliques. At a leaf of any successful execution path of this branching algorithm, we are left with a subgraph H of G that is a cluster graph, and a (possibly reduced) budget $k' \leqslant k$. At this point, we guess the value of ℓ, and extend our solution greedily by: (a) deleting all cliques smaller than ℓ, and (b) for any cliques of size, say q where $q > \ell$, we delete an arbitrary subset of $(q - \ell)$ vertices. We have a valid solution at this if and only if there is some ℓ for which the cost of "uniformizing" the cluster graph H to cliques of size ℓ is within the remaining budget k'.

▶ **Theorem 6.** *2-EIGENVALUE VERTEX DELETION can be solved in time* $\mathcal{O}^\star(3^k)$.

Proof. Let us describe a recursive branching algorithm. Consider an instance, say (G, k), of 2-EIGENVALUE VERTEX DELETION. By Lemma 1, our goal is to decide whether we can delete at most k vertices from G to get a disjoint union of equal-sized cliques. First, we check if G has an induced path on three vertices. This takes polynomial time.

Case 1: G has no induced path on three vertices. The graph G is a disjoint union of cliques, say $C_1, \ldots, C_t$, of sizes $s_1, \ldots, s_t$ respectively. We know that deleting the vertices of any solution results in a disjoint union of equal-sized cliques (say, of size x). Observe that for each $1 \leqslant i \leqslant t$,

- If $s_i \geqslant x$, then $s_i - x$ vertices of the clique C_i are deleted, leaving behind x of its vertices.
- If $s_i < x$, then the entire clique C_i, i.e., all its s_i vertices, are deleted.

So, the overall solution size, i.e., total number of deleted vertices, is

$$\sum_{\substack{1 \leqslant i \leqslant t: \\ s_i \geqslant x}} (s_i - x) + \sum_{\substack{1 \leqslant i \leqslant t: \\ s_i < x}} s_i = \sum_{i=1}^{t} s_i - x \cdot \mu(x)$$

where $\mu(x)$ denotes the number of s_i's amongst $s_1, \ldots, s_t$ such that $s_i \geqslant x$.

Thus, the size of any minimum-sized solution is

$$\sum_{i=1}^{t} s_i - \max_{1 \leqslant j \leqslant t} \left(s_j \cdot \mu(s_j)\right)$$

If this size is $\leqslant k$, we return **YES**; otherwise, we return **NO**. This takes polynomial time.

Case 2: G has an induced path on three vertices, say $a - b - c$. Note that any solution must pick at least one of its three vertices, i.e., a, b, c. So, if $k = 0$, we return NO; otherwise, we guess a vertex that is picked into solution. That is, we branch as follows: In the first (resp. second and third) branch, we include the vertex a (resp. b and c) into solution, delete it from G, and reduce the parameter k by 1. It takes polynomial time to create the subproblems $\big(G \setminus \{a\}, k-1\big)$, $\big(G \setminus \{b\}, k-1\big)$ and $\big(G \setminus \{c\}, k-1\big)$. Next, we run our algorithm on these three instances. If at least one of these three recursive calls returns **YES**, so do we; otherwise, we return **NO**.

The depth of our search tree is at most k. Also, each of its internal nodes has three children. Therefore, it has at most $\mathcal{O}(3^k)$ nodes. Thus, as we spend polynomial time at each node, the overall running time is at most $\mathcal{O}^\star(3^k)$. ◀

We now show that 2-Eigenvalue Vertex Deletion can be solved in polynomial time when the input is a forest. Let (G, k) be an instance of 2-EVD where G is a forest. Note that if S is a valid solution, then $G \setminus S$ is either independent or a disjoint collection of edges.

Therefore, we can arrive at an optimal solution by computing the size of a maximum independent set and a maximum induced matching: this can be done in polynomial time on forests [1, 13]. We also note that a similar argument applies to d-regular graphs for $d \leqslant 2$.

▶ **Proposition 7** ($\star$). *2-Eigenvalue Vertex Deletion admits polynomial time algorithms on forests and d-regular graphs for $d \leqslant 2$.*

3.2 r-EVD for $r \geqslant 3$

To demonstrate the hardness of r-EVD for any fixed $r \geqslant 3$, we give a reduction from Vertex Cover on Cubic Graphs.

▶ **Theorem 8** ($\star$). *Let $r \geqslant 3$ be an integer. Then, r-Eigenvalue Vertex Deletion is NP-complete, even on bipartite graphs of maximum degree four.*

Next, we show that r-EVD is FPT in the combined parmeter $k + \Delta(G)$, where $\Delta(G)$ is the maximum degree of G.

▶ **Theorem 9.** *Let $r \geqslant 3$ be an integer. Then, r-Eigenvalue Vertex Deletion admits an FPT algorithm running in time $\mathcal{O}^\star\Big((r+1)^{2k} \cdot 2^{k^2} \cdot \big(\Delta(G)\big)^{rk}\Big)$.*

Proof Sketch. Let (G, k) be an instance of r-EVD. We claim that if G has more than $(r+1) \cdot 2^k$ eigenvalues, then G is a NO instance, and we can detect this upfront. The intuition is that the Cauchy interlacing structure (Lemma 3) allows us to conclude that one vertex can reduce the number of distinct eigenvalues in the spectrum by a factor of at most half: so if there are "too many" distinct eigenvalues in the spectrum to begin with, k deletions will not suffice to reduce the number of distinct eigenvalues substantially enough. We now quantify this argument: suppose, for the sake of contradiction, that G has more than $(r+1) \cdot 2^k$ eigenvalues, and let $S \subseteq V(G)$ be a subset of at most k vertices such that $A_{G \setminus S}$ has at most r distinct eigenvalues. Denote the vertices of S by $v_1, v_2, \ldots, v_t$, where $t \leqslant k$. By Lemma 3 applied to G, $G \setminus \{v_1\}$, we know that the number of distinct eigenvalues in $G \setminus \{v_1\}$ is at least $\lfloor \frac{1}{2} \eta_G \rfloor$, where η_G is the number of distinct eigenvalues in G. Applying this argument iteratively to $G \setminus \{v_1\}$ and $G \setminus \{v_1, v_2\}$ and so on, it is clear that the number of distinct eigenvalues in $G \setminus S$ is at least $\frac{\eta_G}{2^k} - 1$, but if $\eta_G > (r+1) \cdot 2^k$, then we have a contradiction.

So we assume that G has at most $(r+1) \cdot 2^k$ eigenvalues in its spectrum. Note that if G has a shortest path P with at least r edges then any solution S must contain one of the vertices of P (c.f. Lemma 2). This gives us a branching strategy that can be executed in $\mathcal{O}^\star((r+1)^k)$ time. Let (H, k') be an instance at a leaf of some successful execution path of this branching algorithm. Note that H is a subgraph of G whose diameter is at most $r-1$ and $k' \leqslant k$ is a residual budget.

Let C be a connected component of H. Note that $|C| \leqslant \big(\Delta(G)\big)^r$, in other words, H is a collection of "small" components. Note that if the spectrum of H has more than $(r+1) \cdot 2^{k'}$ eigenvalues, we say NO as before. On the other hand, if the spectrum of H has at most r eigenvalues, then we are already done. So the spectrum of H has more than r and at most $(r+1) \cdot 2^{k'}$ eigenvalues. Otherwise, for the sake of analysis, assume that (H, k') is a **YES**-instance with solution S. Note that there is an eigenvalue λ that belongs to the spectrum of H but not to the spectrum of $H \setminus S$. Note that there is at least one connected component C such that λ belongs to the spectrum of $H[C]$. Therefore, $S \cap C \neq \emptyset$. Our algorithm proceeds by guessing λ and a choice of vertex from $S \cap C$, both of which we can afford because the spectrum of H and the sizes of the components of H are bounded by $(r+1) \cdot 2^{k'}$ and $|C| \leqslant \big(\Delta(G)\big)^r$ respectively. ◀

4 Reducing eigenvalues by adding edges

We show that the 2-Eigenvalue Edge Addition is NP-complete even on cluster graphs, and demonstrate a quadratic kernel in the standard parameter. Also, for any fixed $r \geqslant 3$, we show that the r-EEA problem is NP-complete.

For the first result, we reduce from 3-Partition which is known to be strongly NP-complete [7]. The input for 3-Partition consists of a set $T = \{s_1, \ldots, s_{3n}\}$ and b, where s_i's are positive integers from $\left(\frac{b}{4}, \frac{b}{2}\right)$, s_i's and b are given in unary, and $\sum_{i=1}^{3n} s_i = nb$. The goal of this problem is to decide whether there T can be partitioned into n triplets such that the elements of any triplet sum up to b. The intuition for the reduction is the following: the reduced instance is a disjoint union of cliques whose sizes are $\{s_1, \ldots, s_{3n}\}$ and a large number of cliques of size b. The idea is that a solution to the 3-Partition instance can guide the smaller cliques into appropriate mergers so that all cliques have size b, and the "large" number of cliques of size b, combined with an appropriately chosen budget, essentially forces this solution structure in the reverse direction, allowing us to derive a solution for 3-Partition.

▶ **Theorem 10.** *2-Eigenvalue Edge Addition is NP-complete, even when restricted to cluster graphs, forests, and 2-regular graphs.*

Proof. We describe the hardness for cluster graphs. Consider an instance, say (T, b), of 3-Partition, where $T = \{s_1, \ldots, s_{3n}\}$ such that i) $\frac{b}{4} < s_i < \frac{b}{2}$ for all $1 \leqslant i \leqslant 3n$, and ii) $\sum_{i=1}^{3n} s_i = nb$.

Let us construct a graph, say G, as follows: For every $1 \leqslant i \leqslant 3n$, introduce a clique, say C_i, of size s_i. Also, add $M := 3nb$ cliques, each of size b; let us refer to them as *dummy cliques.* The graph G is the disjoint union of $3n + M$ cliques, namely $C_1, \ldots, C_{3n}$ and the M dummy cliques. Let us show that (T, b) is a **YES** instance of 3-Partition if and only if (G, nb^2) is a **YES** instance of 2-Eigenvalue Edge Addition.

($\Rightarrow$) Suppose that (T, b) is a **YES** instance of 3-Partition. Then, there exists a partition of T into n triplets, say $T = T_1 \uplus \ldots \uplus T_n$, such that for every $1 \leqslant i \leqslant n$, the elements of T_i add up to b. That is, $s_{x_i} + s_{y_i} + s_{z_i} = b$, where $s_{x_i}, s_{y_i}, s_{z_i}$ denote the three elements of T_i.

For every $1 \leqslant i \leqslant n$, merge the three cliques $C_{x_i}, C_{y_i}, C_{z_i}$ into one clique, say D_i, as follows:

- Make every vertex of C_{x_i} adjacent to every vertex of C_{y_i}.
- Make every vertex of C_{x_i} adjacent to every vertex of C_{z_i}.
- Make every vertex of C_{y_i} adjacent to every vertex of C_{z_i}.

See Figure 1 for an illustration.

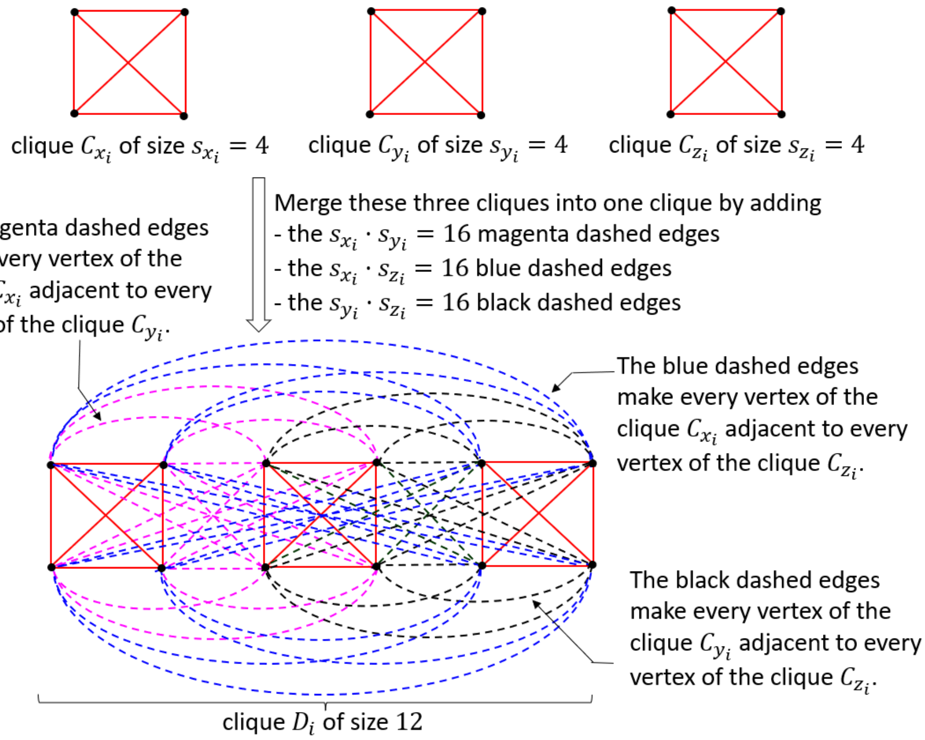


Figure 1 An illustration of the merger of the three cliques $C_{x_i}, C_{y_i}, C_{z_i}$, when $s_{x_i} = s_{y_i} = s_{z_i} = 4$, in Theorem 10.

Note that the number of edges so added to G is

$$\sum_{i=1}^{n} \left(s_{x_i} \cdot s_{y_i} + s_{x_i} \cdot s_{z_i} + s_{y_i} \cdot s_{z_i}\right) < \sum_{i=1}^{n} \left(3 \cdot \frac{b}{2} \cdot \frac{b}{2}\right) < nb^2$$

For each $1 \leqslant i \leqslant n$, the size of the clique D_i is $s_{x_i} + s_{y_i} + s_{z_i} = b$. The resulting graph, say H, is the disjoint union of $n + M$ cliques, each of size b, namely $D_1, \dots, D_n$ and the M dummy cliques. The adjacency matrix of H has two distinct eigenvalues, i.e., -1 and $b - 1$. Thus, (G, nb^2) is a **YES** instance of 2-EIGENVALUE EDGE ADDITION.

($\Leftarrow$) Suppose that (G, nb^2) is a **YES** instance of 2-EIGENVALUE EDGE ADDITION. That is, there exists $S \subseteq \binom{V(G)}{2} \setminus E(G)$ of size $\leqslant nb^2$ such that adding the edges of S to G results in a graph, say H, whose adjacency matrix has at most two distinct eigenvalues. Using Lemma 1, the graph H is a disjoint union of equal-sized cliques. Observe that each clique of H is formed by merging some of the $3n + M$ cliques of G, namely $C_1, \dots, C_{3n}$ and the M b-sized dummy cliques.

First, let us show that no dummy clique participates in a merger. That is, in H, each of the M b-sized dummy cliques of G remains as it is. For the sake of contradiction, assume that there exists a dummy clique that merges with some other clique(s) of G to form a bigger (i.e., of size $> b$) clique of H. Then, as all cliques of H have the same size, none of the other

$M - 1$ b-sized dummy cliques of G can remain as it is. Now, as each of the M b-sized dummy cliques participates in some merger, it is incident to $\geqslant b$ edges of S. Also, every edge of S is incident to at most two dummy cliques. Therefore, we get $|S| \geqslant \frac{Mb}{2} > nb^2$, a contradiction.

Let $D_1, \ldots, D_t$ denote the equal-sized cliques of H other than the M dummy cliques. Note that their common size is the same as that of a dummy clique, i.e., b. Consider any $1 \leqslant i \leqslant t$. The clique D_i is formed by merging some (say p_i) of the $3n$ cliques $C_1, \ldots, C_{3n}$. Each of these p_i cliques has size $> \frac{b}{4}$ and $< \frac{b}{2}$. Also, their sizes add up to the size of the clique D_i, i.e., b. Therefore, we have $p_i \cdot \frac{b}{4} < b < p_i \cdot \frac{b}{2}$. So, we get $p_i = 3$. Hence, each of the t cliques $D_1, \ldots, D_t$ is obtained by merging three of the $3n$ cliques $C_1, \ldots, C_{3n}$. We have $t = n$.

Consider any $1 \leqslant i \leqslant n$. Let $C_{x_i}, C_{y_i}, C_{z_i}$ denote the three cliques amongst $C_1, \ldots, C_{3n}$ whose merger forms the clique D_i. Let T_i denote the triplet that consists of $s_{x_i}, s_{y_i}, s_{z_i}$. As the sizes of the cliques $C_{x_i}, C_{y_i}, C_{z_i}$ add up to the size of the clique D_i, we have $s_{x_i} + s_{y_i} + s_{z_i} = b$. That is, the elements of the triplet T_i add up to b. Thus, as $T = T_1 \uplus \ldots \uplus T_n$, it follows that (T, b) is a **YES** instance of 3-Partition.

In the reduction above, instead of adding a clique on s_i vertices, we could instead add a cycle (resp. path) on s_i vertices, and adjust the budget to account for the missing edges, thereby showing NP-completeness on 2-regular graphs (resp. forests) as well. ◀

Our next result gives a quadratic kernel for 2-Eigenvalue Edge Addition. Let (G, k) be an instance of 2-EEA. We only describe the main intuition of the kernel informally and defer a detailed argument to a full version of this paper. Since we are only allowed to add edges, we "might as well" complete all the connected components of G to cliques and adjust the budget accordingly. Thus, without loss of generality, G is already a cluster graph. Some trivial cases are easily handled, such as: when we cannot afford to complete the original components of G to cliques, or when we have no budget but cliques of different sizes, or when all cliques are already of the same size.

Now, we are left with a situation where we have a non-trivial budget and cliques of at least two distinct sizes. Let the largest sized clique have q vertices, and suppose we have t cliques of size p in G, denoted by $C_1, \ldots, C_t$, where $p < q$. Note that each of these cliques is merged into a larger clique after edges from any valid solution are added to G. In particular, if S is a valid solution, at least $\frac{t \cdot p}{2}$ edges of S are incident to vertices of $C_1 \cup \ldots \cup C_t$. Therefore, if $tp/2 > k$, we can say **NO**. This bounds the sizes of cliques with fewer than q vertices.

For the largest-sized cliques, note that if we have "too many" of them, then none of them are merged into a larger clique after edges from any valid solution are added to G. In particular, it can be shown that if there are s cliques of size q, then if $sq > 2k$, then these cliques are untouched by any valid edge addition set of size at most k. This allows us to throw away most of them, preserving just enough to remember that the cliques must indeed remain untouched in any valid solution. This bounds the number of vertices among the largest sized clique.

Combining these arguments, the overall bound on the total number of vertices in the reduced instance turns out to be quadratic in k. We defer the a detailed proof to a full version of this paper.

▶ **Theorem 11.** *2-Eigenvalue Edge Addition admits a kernel with $\mathcal{O}(k^2)$ vertices.*

Proof. Consider an instance, say (G, k), of 2-Eigenvalue Edge Addition. Owing to Lemma 1, our goal is to decide if we can add $\leqslant k$ edges to G to get a disjoint union of equal-sized cliques. Let us apply the following reduction rules (in the specified order):

Reduction rule 1: Suppose that there's a component, say C, of G, that is not a clique. Then, add the missing $\binom{|V(C)|}{2} - |E(C)|$ edges to turn C into a clique, and reduce the parameter k by $\binom{|V(C)|}{2} - |E(C)|$.

After exhaustively applying Reduction rule 1, G is a disjoint union of cliques; say, it consists of n_1 cliques of size x_1, n_2 cliques of size x_2, ..., n_t cliques of size x_t, where $x_1 < x_2 < \ldots\ldots < x_t$.

Reduction rule 2:

- If $k < 0$, then return **NO**.
- If $k \geqslant 0$ and $t = 1$, then return **YES**.
- If $k = 0$ and $t \geqslant 2$, then return **NO**.

After applying Reduction rule 2, we have $k \geqslant 1$ and $t \geqslant 2$.

Reduction rule 3: If there exists an $1 \leqslant i \leqslant t-1$ such that $n_i \cdot x_i > 2k$, then return **NO**.

Safeness of Reduction rule 3: Suppose that (G, k) is a **YES** instance. Then, there exists $S \subseteq \binom{V(G)}{2} \setminus E(G)$ of size $\leqslant k$ such that adding the edges of S to G results in a disjoint union of equal-sized (say, of size x) cliques. Observe that each of these x-sized cliques is obtained by merging some cliques of G. Note that x is at least the size of a largest clique in G. That is, we have $x \geqslant x_t$. Also, each of the smaller cliques of G, i.e., those of sizes $x_1, \ldots, x_{t-1}$, must participate in some merger.

Now, consider any $1 \leqslant i \leqslant t-1$. Each of the n_i cliques of size x_i is incident to $\geqslant x_i$ edges of S, for it must participate in some merger. Also, any edge of S is incident to at most two of these n_i cliques. Therefore, $|S| \geqslant \frac{n_i \cdot x_i}{2}$. So, as $|S| \leqslant k$, we get $n_i \cdot x_i \leqslant 2k$. Thus, Reduction rule 3 is safe.

After applying Reduction rule 3, we have $n_i \cdot x_i \leqslant 2k$ for all $1 \leqslant i \leqslant t-1$. Also, as $x_1, \ldots, x_{t-1}$ are $t-1$ distinct integers in the interval $[1, 2k]$, we get $t - 1 \leqslant 2k$.

Reduction rule 4: Suppose that $n_t \cdot x_t > 2k$. Then, remove all but $\frac{2k+1}{x_t}$ cliques of size x_t from G.

Safeness of Reduction rule 4: If $n_t \cdot x_t > 2k$, then in any solution, none of the n_t cliques of size x_t participate in a merger. That is, each of them remains as is after the edge additions, and each merger (involving the remaining cliques, i.e., those of sizes $x_1, \ldots, x_{t-1}$) results in an x_t-sized clique. This is because if any clique of size x_t gets to participate in a merger, then each of the remaining $n_t - 1$ cliques of size x_t must also participate in some merger (because all cliques have the same size after the edge additions), thereby needing $\geqslant \frac{n_t \cdot x_t}{2} > k$ edge additions.

Also, we have $n_t \cdot x_t > 2k$ before, as well as after, applying Reduction rule 4. Therefore, it follows that any solution before applying RR4 remains a solution after applying Reduction rule 4, and vice versa. Thus, Reduction rule 4 is safe.

If Reduction rule 4 wasn't invoked, then $n_t \cdot x_t \leqslant 2k$; otherwise, after applying Reduction rule 4, we get $n_t \cdot x_t = 2k + 1$.

Finally, the number of vertices in G is at most

$$n_1 \cdot x_1 + \ldots\ldots + n_{t-1} \cdot x_{t-1} + n_t \cdot x_t \leqslant (t-1) \cdot 2k + (2k+1) \leqslant 4k^2 + 2k + 1.$$

This concludes the proof of Theorem 11. ◀

Next, we show that r-EEA is NP-complete for every fixed $r \geqslant 3$.

▶ **Theorem 12** ($\star$). *Let $r \geqslant 3$ be an integer. Then, r-*Eigenvalue Edge Addition *is NP-complete.*

5 Reducing eigenvalues by deleting edges

In this section, we consider the r-EIGENVALUE EDGE DELETION problem. We defer the NP-completeness of 2-EED to the proof of Theorem 16, where the hardness is implicit. In this section, we present an $\mathcal{O}^*(2^k)$-time FPT algorithm for 2-EED and show that it can be solved in polynomial time on triangle-free graphs. Finally, we prove that r-EED is NP-complete for any fixed $r \geqslant 3$.

The FPT algorithm is similar in spirit to the one we use in the proof of Theorem 6: we branch on induced paths of length three, except we now have a choice of two edges instead of three vertices. In particular, if P is an induced path on $\{a, b, c\}$ with edges $\{a, b\}$ and $\{b, c\}$, we recursively solve the instances $(G \setminus \{a, b\}, k-1)$ and $(G \setminus \{b, c\}, k-1)$.

At the leaves of successful execution paths of this branching algorithm, as before, we have cluster graphs where the cliques are not necessarily of the same size, and a residual budget. Let (H, k') denote such an instance, where H is a subgraph of G consisting of t cliques of sizes $s_1, \ldots, s_t$, and $k' \leqslant k$ is the residual budget. Note that if S is such that $G \setminus S$ is a collection of x-sized cliques for some x, then x must divide each s_i. We show that for an optimal choice of S, x is the GCD of the s_i's. Based on this, it is straightforward to check if the residual budget is sufficient or not.

▶ **Theorem 13.** *2-EIGENVALUE EDGE DELETION admits an algorithm with running time $\mathcal{O}^*(2^k)$.*

Proof. Let us describe a recursive branching algorithm. Consider an instance, say (G, k), of 2-EIGENVALUE EDGE DELETION. Owing to Lemma 1, our goal is to decide whether we can delete at most k edges from G to get a disjoint union of equal-sized cliques. First, we check if G has an induced path on three vertices. This takes polynomial time.

Case 1: G has no induced path on three vertices. The graph G is a disjoint union of cliques, say $C_1, \ldots, C_t$, of sizes $s_1, \ldots, s_t$ respectively. Observe that deleting the edges of any solution breaks each of these t cliques into equal-sized cliques (say, of size x). That is, for every $1 \leqslant i \leqslant t$, it breaks the clique C_i into $\frac{s_i}{x}$ cliques, each of size x. As each of these $\frac{s_i}{x}$ cliques has $\binom{x}{2}$ edges, the number of edges deleted from the clique C_i is

$$\binom{s_i}{2} - \frac{s_i}{x}\binom{x}{2} = \frac{s_i(s_i - x)}{2}$$

So, larger x corresponds to smaller solutions, i.e., fewer edge deletions. Also, x must divide each of $s_1, \ldots, s_t$. Therefore, for any minimum-sized solution, we have $x = \gcd(s_1, \ldots, s_t)$, and its size is

$$\sum_{i=1}^{t} \frac{s_i(s_i - \gcd(s_1, \ldots, s_t))}{2}$$

If this size is at most k, we return **YES**; otherwise, we return **NO**. This takes polynomial time. See Figure 2 for an example.

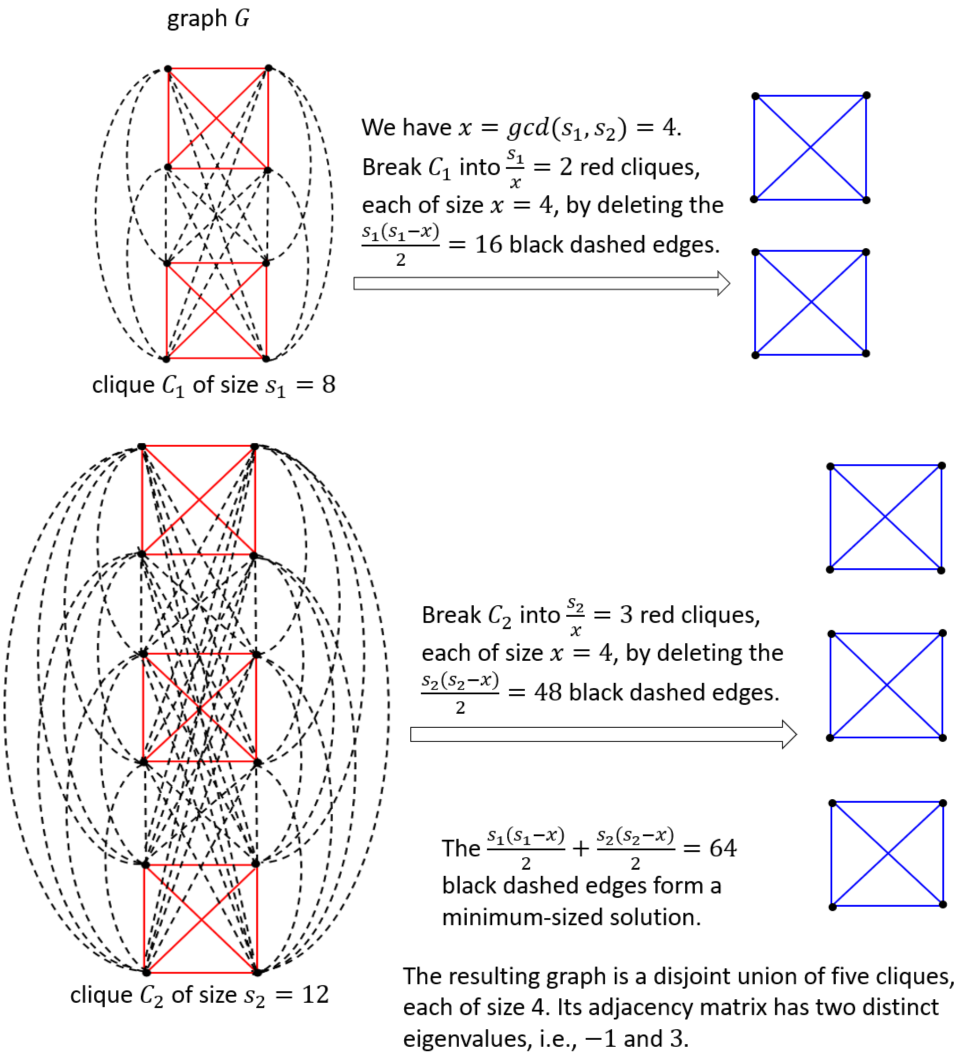


Figure 2 An example illustrating the breaking of cliques in Theorem 13.

Case 2: G has an induced path on three vertices, say $a - b - c$. Note that any solution must pick at least one of its two edges, i.e., $\{a, b\}$ and $\{b, c\}$. So, if $k = 0$, we return NO; otherwise, we guess an edge that is picked into solution. That is, we branch as follows: In the first (resp. second) branch, we include the edge $\{a, b\}$ (resp. $\{b, c\}$) into solution, remove it from G, and reduce the parameter k by 1. It takes polynomial time to create the sub-problems $\big(G - \{a, b\}, k - 1\big)$ and $\big(G - \{b, c\}, k - 1\big)$. Next, we run our algorithm on these two instances. If at least one of these two recursive calls returns **YES**, so do we; otherwise, we return **NO**.

The depth of our search tree is at most k. Also, each of its internal nodes has two children. Therefore, it has at most $\mathcal{O}(2^k)$ nodes. Thus, as we spend polynomial time at each node, the overall running time is at most $\mathcal{O}^\star(2^k)$. This concludes the proof of Theorem 13. ◀

Our next claim takes advantage of the fact that the sizes of the cliques after the removal of any solution is at most two when the input graph is triangle-free and we are only allowed to delete edges. Therefore, the value of the optimal solution is $|E(G)| - |V(G)|/2$ if G has a perfect matching and $|E(G)|$ otherwise. The result follows from the fact that the existence of a perfect matching can be determined in polynomial time [6].

▶ **Proposition 14.** *2-*Eigenvalue Edge Deletion *is polynomial time solvable on triangle-free graphs.*

Now, we show that r-EED is NP-complete by reducing it from Partition into Triangles on graphs of clique number 3 which is known to be NP-complete [4]. The input for Partition into Triangles is a graph G, and the goal is to decide whether $V(G)$ can be partitioned into $\frac{|V(G)|}{3}$ triplets such that every triplet induces a triangle in G.

▶ **Theorem 15.** *Let $r \geqslant 3$ be an integer. Then, r-*Eigenvalue Edge Deletion *is NP-complete.*

Proof. Let us describe a polynomial-time many-one reduction from Partition into Triangles on graphs of clique number 3 to r-Eigenvalue Edge Deletion. Consider an instance, say G, of Partition into Triangles, where G is a graph, say on n vertices and m edges, with clique number 3. Let us construct a graph, say H, from G, as follows: First, we add G as it is. Next, for each $3 \leqslant i \leqslant r+1$, we introduce $M := m-n+1$ cliques, each of size i; let us refer to these cliques as *dummy cliques*. That is, the graph H is the disjoint union of the graph G, M dummy cliques of size 3, $\ldots$, M dummy cliques of size $r+1$. We set the budget to be $m-n$. Let us show that G has $\frac{n}{3}$ pairwise vertex disjoint triangles if and only if $(H, m-n)$ is a YES instance of r-Eigenvalue Edge Deletion.

($\Rightarrow$) Suppose that G has $\frac{n}{3}$ pairwise vertex disjoint triangles, say $T_1, \ldots, T_{n/3}$. Let S denote the set that consists of those $m-n$ edges of G that do not belong to any of these $\frac{n}{3}$ triangles. Note that the graph $H \setminus S$ is the disjoint union of

- $\frac{n}{3}+M$ triangles, namely $T_1, \ldots, T_{n/3}$ and the M dummy cliques of size 3. They contribute two distinct eigenvalues, i.e., -1 and 2.
- M dummy cliques of size 4. They contribute two distinct eigenvalues, i.e., -1 and 3.

  ⋮

  ⋮
- M dummy cliques of size $r+1$. They contribute two distinct eigenvalues, i.e., -1 and r.

So, the adjacency matrix of the graph $H \setminus S$ has r distinct eigenvalues, namely $-1, 2, 3, \ldots\ldots, r$. Thus, $(H, m-n)$ is a **YES** instance of r-Eigenvalue Edge Deletion.

($\Leftarrow$) Suppose that $(H, m-n)$ is a **YES** instance of r-Eigenvalue Edge Deletion. That is, there exists $S \subseteq E(H)$ of size $\leqslant m-n$ such that the adjacency matrix of the graph obtained by deleting the edges of S from H has $\leqslant r$ distinct eigenvalues.

Consider any $3 \leqslant i \leqslant r+1$. Note that the number of i-sized dummy cliques, i.e., M, is $> m-n \geqslant |S|$. So, there's at least one i-sized dummy clique, say C_i, such that none of its edges is deleted. That is, no edge of C_i belongs to S and thus, it appears as a component of the graph $H \setminus S$, thereby contributing two distinct eigenvalues, namely -1 and $i-1$. Thus, it follows that the adjacency matrix of the graph $H \setminus S$ must have $-1, 2, 3, \ldots, r$ as its r distinct eigenvalues.

Now, using Lemma 4, it is clear that the graph $H \setminus S$ must be a disjoint union of some cliques, whose sizes are $3, 4, \ldots, r+1$. So, as G has clique number 3, after removing those edges of G that belong to S, we're left with $\frac{n}{3}$ pairwise vertex-disjoint triangles of G, as desired. This concludes the proof. ◀

6 Reducing eigenvalues by editing edges

In this section, we show that 2-Eigenvalue Edge Editing is NP-complete. We give a reduction from Partition into Triangles.

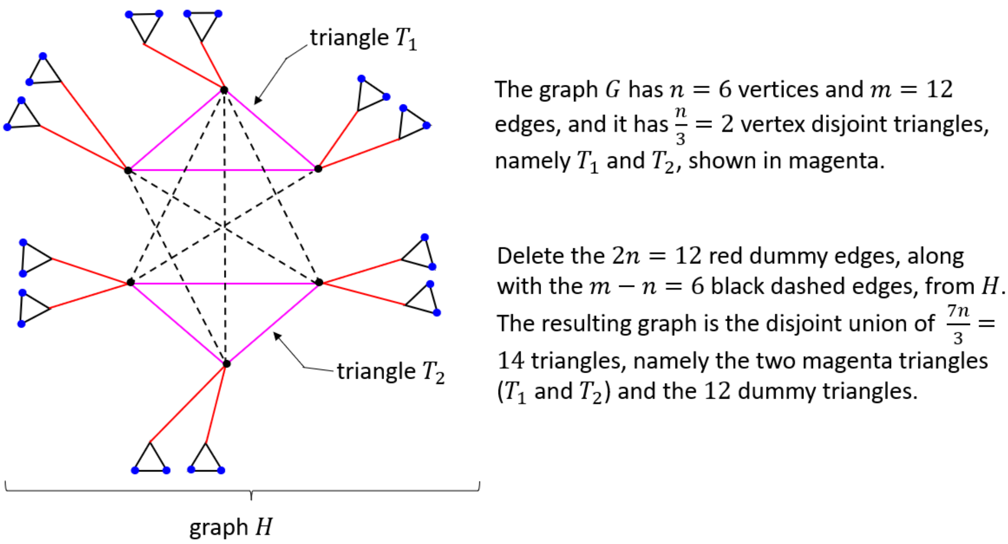
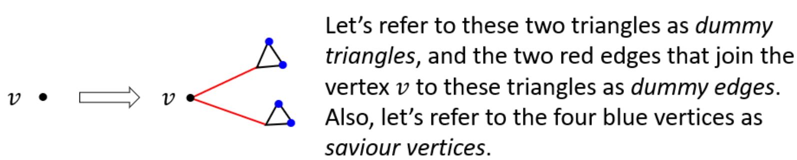


Figure 3 An example illustrating the construction in Theorem 16.

▶ **Theorem 16.** *2-*Eigenvalue Edge Editing *is NP-complete.*

Proof. Let us describe a polynomial-time many-one reduction from Partition into Triangles to 2-Eigenvalue Edge Editing. Consider an instance, say G, of Partition into Triangles, where G is a graph on n vertices and m edges. Let us construct a graph H based on G as follows: for every vertex $v \in V(G)$, attach two triangles to v, as shown below.



See Figure 3 for an illustration.

Note that $|V(H)| = 7n$ and $|E(H)| = m + 8n$. Let us show that G has $\frac{n}{3}$ pairwise vertex disjoint triangles if and only if $(H, m + n)$ is a **YES** instance of 2-Eigenvalue Edge Editing.

($\Rightarrow$) Suppose that G has $\frac{n}{3}$ pairwise vertex disjoint triangles, say $T_1, \ldots, T_{n/3}$. Let $S \subseteq E(H)$ denote the set that consists of the $2n$ dummy edges, along with those $m - n$ edges of G that do not belong to any of these $\frac{n}{3}$ triangles. Note that the graph $H \setminus S$ is the disjoint union of $\frac{7n}{3}$ triangles, namely $T_1, \ldots, T_{n/3}$ and the $2n$ dummy triangles. Its adjacency matrix has two distinct eigenvalues, i.e., -1 and 2. Thus, $(H, m + n)$ is a **YES** instance of 2-Eigenvalue Edge Editing.

($\Leftarrow$): Suppose that $(H, m+n)$ is a **YES** instance of 2-Eigenvalue Edge Editing. That is, there exist $D \subseteq E(H)$ and $A \subseteq \binom{V(H)}{2} \setminus E(H)$ such that: i) $|A| + |D| \leqslant m + n$, and ii) deleting the edges of D from H, and adding the edges of A to H, results in a graph, say H', whose adjacency matrix has at most two distinct eigenvalues. Using Lemma 1, the graph H' is a disjoint union of equal-sized cliques (say, of size x). As each of these $\frac{|V(H)|}{x}$ cliques has $\binom{x}{2}$ edges, the number of edges in H' is

$$\frac{|V(H)|}{x} \cdot \binom{x}{2} = \frac{7n(x-1)}{2}$$

Also, we have $|E(H)| + |A| - |D| = |E(H')|$. Therefore,

$$(m + 8n) + |A| - |D| = \frac{7n(x-1)}{2} \tag{1}$$

Adding (1) to the inequality $|A| + |D| \leqslant m + n$, we get

$$|A| \leqslant \frac{7n(x-3)}{4} \tag{2}$$

Note that each saviour vertex has degrees 2 and $x - 1$ in H and H' respectively. So, each of the $4n$ saviour vertices is incident to $\geqslant x - 3$ added edges $\big($i.e., edges of $A\big)$. Also, any edge of A is incident to at most two saviour vertices. Therefore,

$$|A| \geqslant \frac{4n(x-3)}{2} \tag{3}$$

Using (2) and (3), we get $x = 3$ and $|A| = 0$. Thus, the graph H' is a disjoint union of $\frac{7n}{3}$ triangles, obtained from H by only edge deletions: in other words, no edge additions are involved. This implies that we have $\frac{7n}{3}$ pairwise vertex disjoint triangles, say $T_1, \ldots, T_{7n/3}$, of the $7n$-vertex graph H. Note that the vertices of any dummy triangle belong to a unique triangle $\big($i.e., the dummy triangle itself$\big)$ in H. So, amongst $T_1, \ldots, T_{7n/3}$, we must have the $2n$ dummy triangles. Now, it is clear that the remaining $\frac{7n}{3} - 2n = \frac{n}{3}$ triangles form a collection of pairwise vertex disjoint triangles in G, as desired. ◀

7 Concluding Remarks

We considered the problem of modifying a graph optimally to reduce the number of distinct eigenvalues in the spectrum of its adjacency matrix. These problems turned out to be closely related to, but different from, modifications that aim to reduce the rank of the adjacency matrix and the diameter of the graph.

The complexity of r-EEE for fixed $r \geqslant 3$ remains open. The parameterized complexity of 2-EEE in the standard parameter is open, and the question of finding polynomial kernels for 2-EVD and 2-EED remains open as well. Studying these problems from the perspective of structural parameters or on directed graphs are interesting directions for future work.

References

1. Helmut Alt, Norbert Blum, Kurt Mehlhorn, and Markus Paul. Computing a maximum cardinality matching in a bipartite graph in time o (n1. 5mlog n). *Information Processing Letters*, 37(4):237–240, 1991.
2. Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.
3. Dibyayan Chakraborty, L Sunil Chandran, Sajith Padinhatteeri, and Raji R Pillai. Algorithms and complexity of s-club cluster vertex deletion. In *Combinatorial Algorithms: 32nd International Workshop, IWOCA 2021*, pages 152–164. Springer, 2021.
4. Ante Ćustić, Bettina Klinz, and Gerhard J Woeginger. Geometric versions of the three-dimensional assignment problem under general norms. *Discrete Optimization*, 18:38–55, 2015.
5. Michael Doob. On characterizing certain graphs with four eigenvalues by their spectra. *Linear Algebra and its applications*, 3(4):461–482, 1970.
6. Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.
7. Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

8 Felix Goldberg, Steve Kirkland, Anu Varghese, and Ambat Vijayakumar. On split graphs with four distinct eigenvalues. *Discrete Applied Mathematics*, 277:163–171, 2020.

9 S.M. Meesum, Pranabendu Misra, and Saket Saurabh. Reducing rank of the adjacency matrix by graph modification. *Theoretical Computer Science*, 654:70–79, 2016.

10 Syed M. Meesum and Saket Saurabh. Rank reduction of oriented graphs by vertex and edge deletions. *Algorithmica*, 80(10):2757–2776, 2018.

11 Assaf Natanzon. *Complexity and approximation of some graph modification problems*. University of Tel-Aviv, 1999.

12 Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.

13 Michele Zito. Linear time maximum induced matching algorithm for trees. *Nord. J. Comput.*, 7(1):58, 2000.