Recovery Algorithms for planted structures in Semi-random models

A THESIS

SUBMITTED FOR THE DEGREE OF

Master of Technology (Research)

IN THE

Faculty of Engineering

BY

Rameesh Paul

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

December, 2021

Declaration of Originality

I, **Rameesh Paul**, with SR No. **04-04-00-10-22-19-1-16995** hereby declare that the material presented in the thesis titled

Recovery Algorithms for planted structures in Semi-random models

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2019-2021**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Dr. Anand Louis Advisor Signature

DEDICATED TO

My family and my friends

Acknowledgements

I want to start by expressing my sincere gratitude towards Prof. Anand Louis for mentoring me (sometimes even hand holding) into the world of research. From the day I joined his lab, he has ensured that I see as many nooks and corners of research as possible during my Masters's program here. I would also like to thank him for patiently listening to my preposterous ideas and laboriously explaining why they will not work on numerous occasions. His clarity of thought, depth of understanding, and ability to pull a rabbit out of a hat when generating ideas is something I keenly aspire for.

I would like to deeply thank my collaborators Yash Khanna, Akash Kumar, and Anand Louis, without whom this thesis would not exist. A special thanks to Akash and Anand for always taking out time to answering basic queries and being always up for discussions. I want to thank the amazing faculty members of the Algorithms group here, Siddharth Barman, Satish Govindrajan, Arindam Khan, Anand Louis, and Rahul Saladi. I thank you for the amazing courses, discussions, and advice you offered, which was instrumental in shaping me as a researcher. I would also like to thank faculty members in other departments, Arvind Ayyer, Chirayu Athale, Kunal Chaudhury, Aditya Gopalan, Chandra Murthy, and Himanshu Tyagi, for offering theoretical, proof-based courses which helped me learn skills that were instrumental for my research. I want to thank Ryan O'Donnell, Pravesh Kothari, Lap Chi Lau, Tim Roughgarden, Luca Trevisan, and Yihong Wu for making their graduate-level course offerings publicly available. I want to thank my labmate Sruthi Gorantla for all the discussions and coffee. I would also like to thank the staff at CSA Office for earnestly resolving any issues I faced with a smile on their face.

I want to thank my friends I made during my stay here, Arka Ray and Eklavya Sharma. Through our collaborations and discussions on proofs, algorithms and other apsects of research as doing quality and independent research, I have matured as a researcher. I would like to specifically thank Arka for painstakingly going through this thesis and providing me valuable feedback. I would like to thank my colleagues Utkarsh Joshi, Anand Krishna, Rahul Madhavan, Shravani Patil, Vishakha Patil, Ramakrishna, Virti Savla, and KVN Sreenivas,

Acknowledgements

I want to thank my friends Karan Batta, Ishan Gambhir, Anirudh Garg, Sahil Gupta, Oshin Jain, Kamal Jnagal, Armaan Monga, Aftab Rai, Rishabh Sharma, and Niraj Tanwar for the constant love and support I received from them. Apologies to Anirudh and Niraj for missing your wedding ceremonies owing to my academic commitments. I want to thank my closest friend and partner, Akansha Dimri, for her steadfast support, incessant patience, and constant love, care and affection. I want to thank all my friends for believing in me and their persistent motivation when even I had doubts about myself. I would also like to thank Akansha, Sahil, Karan and Armaan for spending a torrential amount of time and energy on stimulating discussions on topics ranging from technical to health to sports to cinema and life in general.

At last but certainly not least, I would like to thank my mother Sukhwinder Kaur, my father Chaman Lal, and my brother Krishna for their endless love, encouragement, and support. This journey would not have been possible without them.

Abstract

For many NP-hard problems, the analysis of best-known approximation algorithms yield "poor" worst-case guarantees. However, using various heuristics, the problems can be solved (to some extent) in real-life instances. This success can be attributed to the atypicality of worst-case instances in real life, and therefore motivates studying the problem in "easier" instances. Analyzing the problem in *Planted solution models* and *Semi-random models* is one such systematic approach along these lines.

In this thesis, we study *planted solution models* and *semi-random models* for various graph problems. Given a graph G with n vertices, we consider the task of finding the largest induced subgraph of G with a *particular structure*. We start by studying the problem where the *particular structure* is a planar graph. Next, we look at the Odd Cycle Transversal problem or equivalently the problem of finding the largest induced bipartite subgraph. Finally, we study the problem of finding the largest independent set in r-uniform hypergraphs. All these problems are NP-hard and have abysmal worst-case approximation guarantees.

An instance of a *planted solution model* is constructed by starting with a set of vertices V, and choosing a set $S \subseteq V$ of k vertices and adding a *particular structure* on it. Edges between pairs of vertices in $S \times (V \setminus S)$ and $(V \setminus S) \times (V \setminus S)$ are added independently with probability p. The algorithmic task then is to recover this *planted structure*. As a special case for all these problems, when the *planted structure* is an empty graph, the problem reduces to recovering a planted independent set and we don't expect efficient recovery algorithms for $k = o(\sqrt{n})$.

For the problem of finding the largest induced bipartite subgraph, we give an exact recovery algorithm that works for $k = \Omega_p\left(\sqrt{nlogn}\right)$. For the problem of finding maximum independent set in r-uniform hypergraphs, we give an algorithm which works for $k = \Omega_{p,\varepsilon}\left(n^{(r-1)/(r-0.5)}\right)$ and returns an independent set of size $(1-\varepsilon)k$. Our results also hold for a natural *semi-random model* of instances inspired by Feige and Kilian [FK01] model. Our algorithms are based on analyzing continuous relaxations of these problems. We employ techniques from Spectral Graph Theory, Convex Optimization (Linear Programs (LP's) and Semi-Definite Programs (SDP's) relaxations), and Lasserre/Sum-of-Squares hierarchy strengthening of convex relaxations.

Publications based on this Thesis

1. **Independent sets in Semi-random hypergraphs**
   Joint work with Yash Khanna and Anand Louis.
   Algorithms and Data Structures Symposium (WADS 2021)

2. **Exact recovery algorithm for Planted Bipartite Graph in Semi-random Graphs**
   Joint work with Akash Kumar and Anand Louis
   Part of an ongoing manuscript under progress.

Contents

List of Figures

Chapter 1

Introduction

1.1 Beyond Worst-Case Analysis

The traditional worst-case analysis paradigm has various shortcomings. Many optimization problems turn out to be NP-hard in this paradigm. Further, they have "poor" approximation guarantees for the worst-case instances of the problem. However, in practice, many heuristics exist that solve the problem (to some extent) in real-life instances. One could attribute this success to the atypicality of the worst-case instances. This motivates new paradigms for analysis which can give improved guarantees for "easier" instances of the problem.

In an effort to better understand the complexity of various computationally intractable problems, a lot of work is focused on special cases of the problem. However, a more systematic approach is to characterize "easier" instances of the problem and attempt to show that such a characterization holds for real-world instances. There have been various frameworks proposed for studying such "easier" instances as *paramterized algorithms*, *perturbation resilient instances*, *smoothed analysis* etc.

In the area of *parameterized algorithms*, we analyze algorithms with an additional parameter k other than the input size n. We then attempt to show that the problem can be solved exactly in $f(k)n^{O(1)}$ running time (called *fixed parameter tractable*). A useful parameter in graph problems is *treewidth*[1], and many NP-hard problems admit a polynomial-time solution in constant treewidth graphs [Bod88, AP89, FLS+18]. Another approach is to look at the optimal solution as being resilient to a small perturbation in the input instances. For graph problems, one such widely used notion of *perturbation resilience* is *Bilu-Linial stability* [BL12]. In a γ-stable instance, we require that the edge weights can be scaled by a factor of γ and

[1] treewidth is roughly a notion of how close the graph is to a tree. Trees are graphs with treewidth 1.

this perturbed instance still has the same optimal solution[2]. In the work [MMV14a], they show that for $\mathcal{O}\left(\sqrt{\log n}\log\log n\right)$ stable instances of the MAX-CUT problem, a standard SDP relaxation is integral. They also give an LP relaxation which is exact for 4-stable instances of min multiway cut problem.

Another notion of worst case analysis is that of *smoothed analysis* where one starts with a worst-case instance of a problem and considers bounded random perturbations to the input instance. Spielman and Tang [ST01] analyzed the simplex algorithm and argue its effectiveness under this analysis framework. In a recent work, Makarychev and Makarychev [MM20] extend the notion of smoothed analysis by defining a notion of *certified algorithms*. In certified algorithms, they allow an input instance to perturb itself (need not be bounded and random), and the algorithm returns an optimal solution (certificate) to the perturbed instance.

Another systematic approach (this is the one we consider in this thesis) is to study the problem in various *random* and *semi-random* models. Here, one starts with solving the problem for random instances (for graph problems, this is often $G_{n,p}$ Erdős-Rényi graphs). In an Erdős-Rényi graph, for each pair of vertices, an edge is added independently with probability p. The analysis in random instances is often much simpler, and one can give algorithms with "good" approximation guarantees. As a running example, we consider the largest independent set/clique problem. The worst-case instance of the problem has an inapproximability result of $n^{1-\varepsilon}$ for any $\varepsilon > 0$. However in $G_{n,p}$ model, the size of largest clique can be shown to be close to $(2 \pm o(1))\log_{1/(1-p)} n$ with high probability. A simple greedy algorithm [GM75], which considers vertices in an arbitrary order and tries to add them to "clique so far", can be analyzed to show that it gives a 2-approximation with high probability[3]. Curiously, improving upon this factor of 2 is a major open problem [Rou21].

The next goal in this direction is to plant a solution that is "clearly optimal" in an ambient random graph and then given an algorithm to recover this planted solution. We, therefore, build towards the worst-case instances of the problem by progressively weakening our assumptions. This is called *planted solution model* or *random planted model*. In the planted clique/independent set problem we plant a clique/independent set of size k in an otherwise random $G_{n,p}$ graph. By planting a clique we mean that the we select an arbitrary set of vertices S and add an edge between every pair of vertices in $S \times S$. The work [AKS98] presents an algorithm, which, given a graph $G \sim \mathcal{G}_{n,1/2}$ with a planted clique/independent set of size k, recovers the planted clique when $k > c_1\sqrt{n}$ (where c_1 is a constant). Such planted solution

[2]A weaker notion of stability (γ, δ)-stable allows the new optimal to be within a factor of δ from the optimal value of the original instance.

[3]Since there is a non-zero probability of "hard" instance being generated by such probabilistic model, the guarantees will always be of this form.

models have been studied in the context of other problems as well. We study our problems in a similar *planted solution model* defined as follows.

Definition 1.1 (Planted solution model) *Given n, k, p, our graph with planted structure is constructed as follows,*

1. *Let V be a set of n vertices. Fix an arbitrary subset $S \subset V$ such that $|S| = k$.*
2. *Add edges arbitrarily in the graph induced by S such that it has the planted structure.*
3. *For each pair of vertices in $S \times (V \setminus S)$, add an edge independently with probability p.*
4. *For each pair of vertices in $(V \setminus S) \times (V \setminus S)$, add an edge independently with probability p.*

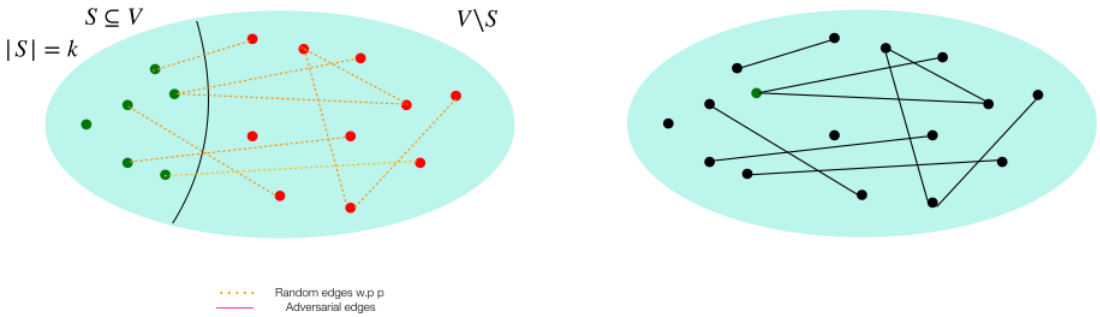

Figure 1.1: Planted solution model Definition 1.1 (left) and Algorithm's view (right).

These models, however, are not realistic since real-world graphs are not random. The algorithms developed in such models are not robust and have little hope of generalizing to real-world instances. Models stronger than planted solution models have also been considered for the clique problem. A key idea towards making these models robust was proposed by Blum and Spencer [BS95], which is to allow an adversary to modify the graph. The adversary's actions have to be limited since otherwise, it can convert the graph to a worst-case instance of the problem. A reasonable assumption is that the adversary should respect the planted solution. This is accomplished by allowing only monotone adversaries, which can either add or delete edges but not both. At the outset, these may seem to be more like allies, but the analysis for algorithms based on degree concentration [Kuc95] and spectral algorithms [AKS98] is not

known to work under presence of such adversaries. In this way, such adversaries encourage the design of robust algorithms that could generalize to real-world instances. For the independent set problem, the work [FK00] showed independent set can still be recovered for $k = \Omega(\sqrt{n})$ under the presence of a certain kind of an adversary. However, this now requires to appeal to the power of semidefinite programming (SDP). The [FK00] model also called as the *sandwich model*, is one such adversarial model and is defined as,

Definition 1.2 (Sandwich model) *Given n, k, p, our graph with planted structure is constructed as follows,*

1. *Let V be a set of n vertices. Fix an arbitrary subset $S \subset V$ such that $|S| = k$.*
2. *Add edges arbitrarily in the graph induced by S such that it has the planted structure.*
3. *For each pair of vertices in $S \times (V \setminus S)$, add an edge independently with probability p.*
4. *For each pair of vertices in $(V \setminus S) \times (V \setminus S)$, add an edge independently with probability p.*
5. *Allow a monotone adversary to add edges between pairs of vertices in $(V \setminus S) \times (V \setminus S)$ and $S \times (V \setminus S)$.*

The action of *monotone adversary* has to be defined appropriately such that the planted solution remains optimal, e.g., for the independent set problem, the monotone adversary is only allowed to add edges.

The work [FK01] introduced a strong *semi-random* model (referred to as the *Fiege and Kilian* model), and gave recovery algorithms for various problems in this model.

Definition 1.3 (Feige-Kilian [FK01] model) *Given n, k, p, our graph with planted structure is constructed as follows,*

1. *Let V be a set of n vertices. Fix an arbitrary subset $S \subset V$ such that $|S| = k$.*
2. *Add edges arbitrarily in the graph induced by S such that it has the planted structure.*
3. *For each pair of vertices in $S \times (V \setminus S)$, add an edge independently with probability p.*
4. *Allow a monotone adversary to add edges between pairs of vertices in $(V \setminus S) \times (V \setminus S)$ and $S \times (V \setminus S)$.*

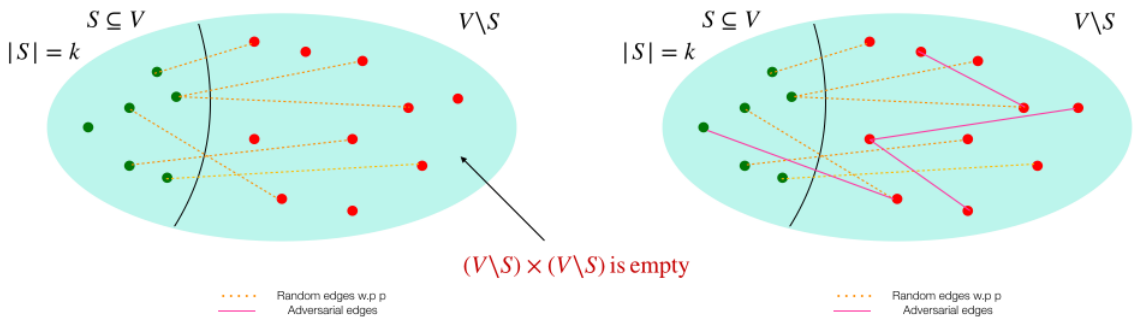

Figure 1.2: Constructing [FK01] model for Independent set problem.

The key point to note here is that $(V \setminus S) \times (V \setminus S)$, a large portion of the graph, is entirely under the control of an adversary. In fact, the way we have defined recovery, it is information-theoretically impossible to recover the planted structure for $k \leq n/2$. However, as we will see, for $k = o(n)$ we can still solve the problem by some weak and reasonable assumptions on $V \setminus S$ or by relaxing our notion of recovery. The work [MMT20] shows that one can recover the planted clique for $k = \Omega_p\left(n^{2/3}\right)$[4] where they relax the notion of recovery and allow an algorithm to output n independent sets instead of one, and give high probability guarantees that at least one of them is the planted set. We also study our problems (discussed in Section 1.3) in [FK01] model and semi-random models inspired by the [FK01] model.

For planted cliques, a lot of work has been done in the special case of $p = 1/2$. However, people have studied other problems such as the planted bisection problems [FK01], and exact recovery problems in SBM [ABH16] in $p = o(1)$ regimes. Therefore, for the problems we consider, we also aim to solve in $p = o(1)$ regimes. We refer to the book [Rou21] for a more detailed discussion of these models in the context of other problems like planted clique, planted bisection, k-coloring, Stochastic Block Models, and Matrix completion problems.

1.2 Graph problems in semi-random models

A wide variety of random graph models and their relaxations have been a rich source of algorithmic problems on graphs.

Coloring problems: Alon and Kahale [AK97] sharpened the results of Blum and Spencer [BS95] and gave algorithms that recover a planted 3-coloring in a natural family of random 3-

[4] Ω_p hides $\mathsf{poly}\,(1/p)$ factors.

colorable instances. The work [FK01] combines the SDP from [AK97] with hyperplane rounding and solves the problem for k-colorable graphs (where k is a constant) and a large range of p in the [FK01] model. The work [CO07] extends this to $k = \mathcal{O}(\sqrt{n})$ and a large range of p in a similar semi-random model. The work [KLT17] showed how to recover a 3-coloring when the input graph is pseudorandom (has some mild expansion properties) and is known to admit a random like 3-coloring. In the work [DF16], they propose a *hosted coloring framework* which generalizes other semi-random models by considering random/adversarial choices for the host graph and planted solution. They extend the [AK97] algorithm to work even when the host graph is a d-regular expander graph.

Independent set/planted clique problem: The *Feige-Kilian model* [FK01] is one of the strongest semi-random models. In [FK01], they also give recovery algorithms for planted clique, planted k-colorable, and planted bisection problem in this model. In [MMT20], they give a recovery algorithm for the independent set problem for $k = \Omega_p\left(n^{2/3}\right)$ and large range of p. The work [KLP21] generalizes these results to r-uniform hypergraphs for $k = \Omega_p\left(n^{(r-1)/(r-0.5)}\right)$ and a large range of p in this model.

Graph partitioning problems: There are other works that study graph partitioning problems in various random and semi-random models. Recovering k planted clusters/communities in a random graph is a popular graph partitioning model also called Stochastic Block Models (SBM). Starting from the work [HLL83], there has been a large body of work studying algorithms for these models [Bop87, SN97, MNS12, ABH16, CZ20], to state a few. For $k = 2$, the problem is known as planted bisection model, which was also studied in the work [FK01] in a semi-random model.

The works [MMV12, MMV14b] study a variety of graph partitioning problems such as Multi cut, Sparsest cut, Balanced cut, Uncut, and Small Set Expansion in a semi-random model equivalent to the [FK01] model. They also give bi-criteria approximation algorithms for a large range of parameters. The works by Louis and Venkat [LV18, LV19] study the problem of balanced vertex expansion and balanced k-way edge expansion in semi-random models.

Densest k-subgraph problem A host of work has been done in various random and semi-random models for the more general densest k-subgraph problem. The works by Hajek, Wu, and Xu [HWX16a, HWX16b, HWX16c] study the problem when the planted dense subgraph is random and gives algorithms (in some range of parameters) for exact recovery using SDP relaxations. They complement these results by providing information-theoretic limits for regimes where recovery is impossible. The work by [BCC+10] studies this problem when the planted graph is arbitrary. They analyze an SDP-based method to distinguish the dense graphs from

the family of $G_{n,p}$ graphs when $k \geq \sqrt{n}$. The work [KL20] studies the problem of d-regular densest k-subgraph in some semi-random model and gives a partial recovery algorithm for some regimes of d, k, n, p.

1.3 Our Contributions

In this thesis, we study three problems where the task is of finding the largest induced subgraph in a given graph. We study these problems in *planted solution model*, *sandwich model* and *semi-random models*.

- **Largest induced planar subgraph problem**
  Given a graph $G = (V, E)$, the problem asks to find the largest induced planar subgraph of G. We study this problem in the planted solution model constructed as per Definition 1.1 where the planted structure is an arbitrary planar graph. We give an algorithm which returns a list of $n^{O(1/p)}$ sets, one of which is exactly the planted planar graph in this model for $k = \Omega_p(\sqrt{n})$.

- **Odd Cycle Transversal (largest induced bipartite subgraph) problem**
  Given a graph $G = (V, E)$, the problem asks to find the largest induced bipartite subgraph of G. We start by studying the problem in planted solution model as per Definition 1.1 where the planted structure is a d-regular balanced bipartite subgraph. For $k = \Omega_p(\sqrt{n})$ and a large range of p, we give an algorithm that recovers the planted bipartite graph exactly w.h.p. The running time of this algorithm is exponential in the number of small eigenvalues (smaller than $-d/2 + 2\sqrt{n}$) of the adjacency matrix of the graph induced on S.

  For many special classes of instances such as, (i) when the probability $p = \Omega(1)$, (ii) when the planted graph is a complete bipartite graph (this is called the balanced biclique problem), (iii) when the planted bipartite graph is random or (iv) more generally when the planted graph is a d-regular expander graph; the number of these small eigenvalues is a constant. Therefore, this already gives us a polynomial-time algorithm.

  However, we attempt to develop an algorithm for an arbitrary d-regular bipartite graph in a semi-random model inspired by the [FK01] model. Our semi-random model (refer Chapter 3 for details) also captures the sandwich model and the planted solution model. For $k = \Omega_p\left(\sqrt{n \log n}\right)$ and a large range of p, we give a polynomial time algorithm that recovers the planted bipartite graph exactly w.h.p.

- **Maximum independent set in r-uniform hypergraphs**

Given a r-uniform hypergraph, the problem asks to find the largest independent set (a set where no hyperedge is completely contained inside the set). We study the problem in [FK01] model with the notion of recovery as per [MMT20] where they output an independent set of size $(1-\varepsilon)\,k$ for any $\varepsilon \in (0,1)$. For $k = \Omega_p\left(\frac{n^{(r-1)/(r-0.5)}}{\varepsilon^{1/(r-0.5)}}\right)$ and a large range of p, we give a polynomial time algorithm that outputs an independent set of size $(1-\varepsilon)\,k$ w.h.p. The result here is the generalization of analogous results for graphs given by [MMT20].

We note that for all these problems, the worst-case instance of the problem is NP-hard. This follows from a seminal result by [Yan78], which shows that for a broad class of graph problems that have a structure which is *hereditary* on induced subgraphs; the problem of finding such a structure is NP-Complete. A graph property is called hereditary if it is inherited by all induced subgraphs. Therefore, planarity, bipartiteness, and independent set are hereditary properties, and hence the respective problems are NP-hard. We discuss further intractability for these problems in relevant chapters.

We note that as a special case to these problems when the planted structure is an empty graph[5] (has no edges), the problem reduces to recovering a planted independent set and we do not expect efficient algorithms for $k = o(\sqrt{n})$ due to [FGR+13, BHK+16].

1.4 Organisation

In Chapter 2, we provide relevant background on various techniques used in this thesis. In Chapter 3, we study the Odd Cycle Transversal or equivalently the largest induced bipartite subgraph problem in planted solution model, sandwich model and a semi-random model inspired by [FK01]. Chapter 4 discusses the problem of finding the largest independent set in r-uniform hypergraphs where the hypergraph is constructed as per [FK01] model. In Chapter 5, we discuss the problem of the largest induced planar subgraph and study it in the planted solution model. Finally, we conclude with some open problems in Chapter 6.

[5]The empty graph is a valid planar graph, a valid bipartite graph as well as a valid independent set

Chapter 2

Preliminaries

In this chapter, we give a brief discussion on the tools typically used for tackling problems in this area. We start with establishing the notation in Section 2.1 and some relevant facts from linear algebra and probability in Section 2.2. In Section 2.3, we present some fundamental results in the Perturbation Theory of matrices. We will focus on symmetric matrices since the adjacency matrix arising out of graphs[1] we study are symmetric. Also, we will mention some fundamental results for random matrices. In Section 2.4, we touch upon various aspects of Semidefinite programs. We discuss how an SDP relaxation is related to a vector relaxation of the program, also called the Vector program (VP). We give another viewpoint of SDP as an LP trying to enforce moment constraints, and this viewpoint can be generalized to higher degree terms and thus achieving higher-order SDP relaxation. This is known as the Lasserre/Sum-of-squares(SOS) hierarchy relaxation, and we discuss it in its own right in Section 2.4.3.

We note that the discussion of these topics is not comprehensive and is only intended to aid understanding of the usage of these tools in upcoming chapters. For a detailed and more exhaustive discussion on these topics, we refer the reader to standard textbooks in respective areas. Our treatment in this chapter for Perturbation Theory is based on the book [Ver18], for SDP is based on the books [BV04, WS11], and for Lasserre/SOS hierarchy, we refer to the survey [Rot13] and the monograph [FKP19]. We start with the following notation used throughout this thesis.

2.1 Notation

- $[M]_{n\times n}$ denotes a matrix M of size $n \times n$. For some set of indices $R_1, R_2 \subseteq [n]$, $M_{R_1\times R_2}$ denotes a matrix of size $n \times n$ constructed out of matrix M of size $n \times n$ by copying the

[1] We will always assume, unless explicitly stated, that the graph is undirected.

entries for $(i,j) \in R_1 \times R_2$ and setting rest of the entries to be 0.

- $M|_{R_1 \times R_2}$ denotes the matrix of size $|R_1| \times |R_2|$ constructed from a matrix M of size $n \times n$ by taking rows corresponding to R_1 and columns corresponding to R_2.
- $E(U,V)$ for some sets U, V denotes the set of edges going between the sets U and V.
- $N(i)$ denotes the set of vertices in the neighborhood of the vertex i.
- All vectors are denoted with a bold typeface.
- For vectors $\mathbf{x}, \mathbf{y}$ when we say $\mathbf{x} \geq \mathbf{y}$ we mean entry wise $x_i \geq y_i$.
- $\mathbb{1}$ denotes a vector which has 1 in all it's entries and $\mathbb{1}_S$ denotes the support/indicator vector where the i^{th} is 1 for $i \in S$ and 0 otherwise.
- For any matrix A, the eigenvalues of A are ordered as $\lambda_1(A) \leq \lambda_2(A) \leq \ldots \leq \lambda_n(A)$. We will drop the matrix A when it is clear from the context. Also the eigenvectors denoted as $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ will assumed to be sorted by their corresponding eigenvalue.
- $\langle A, B \rangle$ is denotes the standard inner product of two matrices of same size $A_{n\times n}$ and $B_{n\times n}$ and $\langle A, B \rangle = \sum_{1 \leq i,j \leq n} A_{ij} B_{ij}$.

2.2 Linear Algebra and Probability

We state a few useful facts from Linear Algebra and probability that will be useful to review for upcoming chapters of this thesis.

2.2.1 Linear Algebraic Facts

Fact 2.1 (Variational characterstic of eigenvalues) *For a given symmetric matrix* $[A]_{n\times n}$,

$$\lambda_1 = \min_{\mathbf{x} \in \mathbb{R}^n} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \text{ and } \lambda_n = \max_{\mathbf{x} \in \mathbb{R}^n} \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

The term $\left(\mathbf{x}^T A \mathbf{x}\right) / \mathbf{x}^T \mathbf{x}$ is also called the Rayleigh quotient of a vector $\mathbf{x}$ with matrix A. The spectral norm of a matrix (denote by $\|A\|_2$) is defined as,

$$\|A\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \max\{|\lambda_1|, |\lambda_n|\}$$

2.2.2 Probabilistic Inequalities

Fact 2.2 (Hoeffding Bound, Theorem 4.14 - [MU17]) *Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{E}[X_i] = \mu_i$ and $\mathbb{P}[a_i \leq X_i \leq b_i] = 1$ for constants a_i and b_i. Then,*

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mu_i\right| \geq \varepsilon\right] \leq 2\exp\left(\frac{-2\varepsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$

Fact 2.3 (Chernoff bound (Multiplicative); Theorem 4.4 (Part 2) - [MU17]) *Consider $X_1, X_2, \ldots, X_n$ be i.i.d. bernoulli variables such that $\mu = \mathbb{E}[\sum_{i=1}^{n} X_i]$. Then for any $\delta \in (0, 1)$,*

$$\mathbb{P}\left[\sum_{i=1}^{n} X_i \leq (1-\delta)\mu\right] \leq \exp\left(-\frac{\mu\delta^2}{2}\right).$$

2.3 Perturbation Theory

Given two symmetric matrices A and B, a fundamental result which is considered as a triangle inequality for matrices, relates the eigenvalues of A and $A + B$ as,

Fact 2.4 (Weyl's inequality) *Let A and B be $n \times n$ symmetric matrices with eigenvalues denoted by $\lambda_1(A), \ldots, \lambda_n(A)$ and $\lambda_1(B), \ldots, \lambda_n(B)$ respectively, then the eigenvalues of $A + B$ (denoted by $\lambda_i(A + B), \forall i \in [n]$) are bounded as,*

$$\lambda_i(A) + \lambda_1(B) \leq \lambda_i(A + B) \leq \lambda_i(A) + \lambda_n(B).$$

However, often it is of more interest as how the eigenvectors of A behave under the effect of a perturbing matrix[2] B. This can be studied using the framework of the Davis-Kahan theorem, originally proved in the work [DK70]; however, we will use the version presented in Theorem 4.5.5 in [Ver18].

Fact 2.5 (Davis-Kahan Theorem) *Let A be an $n \times n$ symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ respectively. Let B be another $n \times n$ symmetric matrix. Consider $M = A + B$ and let it's eigenvectors be $\mathbf{v}_1, \ldots, \mathbf{v}_n$ respectively. Let θ_i be the smaller angle between the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ then,*

$$\sin\theta_i \leq \frac{2\|B\|}{\min_{j \neq i}|\lambda_i - \lambda_j|}.$$

[2] We will use the expression "perturbation matrix" to qualitatively express that a matrix is considered/desired to be small e.g have a small norm

For our model in Definition 1.1, we can express it's adjacency matrix A as,

$$\begin{aligned} A &= A_{S\times S} + (A - A_{S\times S}) \\ &= A_{S\times S} + \mathbb{E}[A - A_{S\times S}] + R \qquad (\text{ We define } R_{ij} = A_{ij} - \mathbb{E}[A_{ij}]) \\ &= A_{S\times S} + p\left(\mathbb{1}\mathbb{1}^T - \mathbb{1}_S\mathbb{1}_S^T\right) + R \end{aligned} \tag{2.1}$$

where $A_{S\times S}$ represents the matrix corresponding to the planted graph. The term $p\left(\mathbb{1}\mathbb{1}^T - \mathbb{1}_S\mathbb{1}_S^T\right)$ is the expected adjacency matrix corresponding to $A - A_{S\times S}$, since in our model we have an edge independently with probability p. The way we define R in eq. (2.1), it's a perturbation matrix for the matrix corresponding to the random part of graph. Since our other models (discussed in Section 1.1) are also built upon random graphs, the family of random matrices is of particular interest to us. The work [Wig58] studied the distribution of eigenvalues for random matrices. Here we present a fundamental result which is informally called the Wigner's theorem; we will use a similar version from the work [Vu07]. Let A denote the adjacency matrix of the resulting graph.

Claim 2.1 *For the matrix R as defined in eq. (2.1) we have $\|R\| \leq 2\sqrt{n}$ holds almost surely.*

Proof: R is symmetric random matrix, and the entries R_{ij} are given as $R_{ij} = A_{ij} - \mathbb{E}[A_{ij}]$. R_{ij}'s can be treated as independent[3] random variables which are bounded between -1 and 1, that have expected value 0 and variance $p(1-p) \leq 1/4$. Now using hence by Theorem 1.1 in the work [Vu07] we have $\|R\| \leq 2\sqrt{n}$ holds almost surely. □

Remark 2.1 *If the perturbation matrix B in Weyl s inequality (Fact 2.4) is a random matrix (corresponding to a random graph), the eigenvalues are shifted only by $\pm 2\sqrt{n}$ i.e $\lambda_i(A+B) \in [\lambda_i(A) - 2\sqrt{n}, \lambda_i(A) + 2\sqrt{n}]$ almost surely.*

2.4 Semidefinite Programming (SDP)

2.4.1 Different facets of SDPs

Conic programs and Semidefinite programs: In this section we will discuss Semidefinite programs (SDP), a broad class of convex optimization problems. A convex optimization problem is an optimization problem which consists of a convex objective function to be optimized over a set of convex constraints (called the feasible set). Perhaps, the most well known class of convex

[3] Since A_{ij}'s were independent in our model.

optimization problems is a Linear Program (LP), where the task is to optimize a linear function over linear inequalities (that form polytope as the feasible convex set).

A more general and useful class of convex programs is a conic program which can be represented (in vector form) as,

Conic Program 2.1

$$\min \mathbf{c}^T \mathbf{x}$$

subject to

$$A\mathbf{x} = \mathbf{b} \tag{2.2}$$

$$\mathbf{x} \in \mathcal{K}\,. \tag{2.3}$$

where the objective function is linear but the feasible set $\mathcal{K}$ is a convex cone. We recall that a set $\mathcal{K}$ is called a convex cone if

$$\alpha_1 \mathbf{x}_1 + \alpha_1 \mathbf{x}_1 \in \mathcal{K}, \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{K} \text{ and } \forall \alpha_1, \alpha_2 \geq 0\,.$$

SDP is a special class of conic programs where the convex cone $\mathcal{K}$ is a set of positive semidefinite matrices[4]. SDP's capture a variety of constraints including linear inequalities, inequalities with quadratic terms and second order cone constraints.

SDP in Combinatorial optimization: In combinatorial optimization, we are interested in integer programs and therefore SDP arise as a natural convex relaxations to these integer programs. As an example, we consider the MAX-CUT problem, where given a graph $G = (V, E)$ the objective is to find a cut $(S, V \setminus S)$ that maximizes the number of edges cut. The MAX-CUT problem can be naturally expressed as an integer quadratic program,

QP 2.1

$$\max \frac{\sum_{\{i,j\} \in E} (x_i - x_j)^2}{4}$$

subject to

$$x_i \in \{-1, 1\}\ . \tag{2.4}$$

Using $X = \mathbf{x}\mathbf{x}^T$, the quadratic program can be reframed as,

[4]The geometric shape of the cone of SDP matrices is known as spectrahedron in optimization literature

SDP 2.1 *(rank constrained non-convex SDP form)*

$$\max \frac{\langle L, X \rangle}{4}$$

subject to

$$X \succeq 0 \tag{2.5}$$

$$\langle X, B_i \rangle = 1, \quad \forall i \in [n] \tag{2.6}$$

$$\mathsf{rank}\,(X) = 1\,. \tag{2.7}$$

where B_i is a matrix which is 1 in (i, i) entry and 0 elsewhere, and L is the Laplacian matrix of the graph, i.e, $L = D - A$ where D is a diagonal matrix having entry D_{ii} as the degree of vertex i and A is the adjacency matrix of the graph. We note that as written above, SDP 2.1 is just a reformulation of QP 2.1 using matrix entries as decision variables. Formally, this is not an SDP because of the non-convex rank constraint[5] which makes solving such a problem NP-hard in general.

SDP in matrix form: Now dropping the non-convex rank constraint, we obtain a set of p.s.d matrices as a feasible set and these form a convex cone. The resulting program is called an SDP in matrix form (dual of the form in Conic Program 2.1). Next, we present a geometric viewpoint, where we consider a different way to relax the same quadratic program.

Vector Programs: Another way to relax QP 2.1 above, is by relaxing x_i to be a d-dimensional vector $\mathbf{x}_i$. The product terms $x_i x_j$ can be written as inner product/norms. Thus, for our MAX-CUT QP 2.1 we obtain a vector program as,

VP 2.1

$$\max \frac{1}{4} \sum_{\{i,j\} \in E} (1 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle)$$

subject to

$$\|\mathbf{x}_i\|^2 = 1, \forall i \in V\,. \tag{2.8}$$

Now we replace the term $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ by a place holder variable y_{ij}. Thus we obtain a LP (linear program) in these new variables y_{ij}'s, which can be efficiently solved. However, as such, this would be a relaxation of a relaxation since we didn't enforce the y_{ij}'s to be of the form $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

[5] In optimization literature these are called rank constrained SDPs.

To enforce this, we arrange the variables y_{ij}'s in a matrix $Y = [y_{ij}]_{n\times n}$ and add the constraint that $Y = XX^T$ where X is an $n \times d$ matrix with rows as vectors $\mathbf{x}_i$'s. We recognize that the form $Y = XX^T$ is a characterization of p.s.d. matrices, and hence one obtains an SDP of form SDP 2.1 without the rank constraint.

We are not done yet, since $\mathsf{rank}\,(Y) = \min\{n, d\}$ and hence for $d < n$, we need to add this rank constraint. However, this can be avoided if we choose $d \geq n$ i.e. relax x_i to n-dimensional vector. Therefore, it shows that the convex program we obtain by relaxing x_i to n-dimensional vectors is equivalent to our earlier relaxation of SDP in matrix viewpoint.

Generalizing from this MAX-CUT example, formally we write a Vector Program as a linear program over the dot product as,

VP 2.2

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

subject to

$$\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} \langle \mathbf{x}_i, \mathbf{x}_j \rangle = b \tag{2.9}$$

$$A, C \succeq 0\,. \tag{2.10}$$

Moment viewpoint A yet another way of relaxing the quadratic program of the form QP 2.1 is by replacing the terms $x_i x_j$ by a single term y_{ij}. This way one obtains an LP in the variables y_{ij}'s but again we need to enforce the constraint that $y_{ij} = x_i x_j$. This constraint is called the moment constraint and here the relaxation idea is to try and enforce this non-linear moment constraint by a system of linear constraints. This can be systematically done by considering that

$$\left(\sum c_i x_i\right)^2 \geq 0, \forall c_i \in \mathbb{R} \tag{2.11}$$

and we replace the occurrence of the product terms $x_i x_j$ in eq. (2.11) by y_{ij}. Once again arranging y_{ij}'s in a matrix Y, the condition above can be rewritten as $\mathbf{c}^T Y \mathbf{c} \geq 0$ for all $\mathbf{c} \in \mathbb{R}^n$. We recall that this is a yet another characterization of p.s.d. matrices. Therefore, we again obtain an SDP of the form SDP 2.1 without the rank constraint.

It is also evident from this form that why an SDP can be solved efficiently. It is because after the relaxation, we is just have an LP. Further, the ellipsoid algorithm for solving LP's requires a separation oracle when a constraint is violated. This separation oracle is in the

form of a hyperplane that separates the feasible region from the region where the constraint is valid. If the constraints were themselves linear (as in an LP), the separation oracle would be the violating constraint itself. For the set of p.s.d matrices, the vector $\mathbf{c}$ (which has to exist) such that $\mathbf{c}^T Y \mathbf{c} < 0$ acts as such a hyperplane. In fact, since the eigenvalues of a p.s.d matrix are all non-negative, a violating constraint means a negative eigenvalue. The eigenvector corresponding to this negative eigenvalue gives us a separation oracle for non-p.s.d. matrices.

2.4.2 SDP Duality

Similar to Linear Program(LP), taking the dual of an SDP is a useful technique that provides an explicit certificate for optimality. The dual program is computed by taking the Lagrangian and computing the Lagrange dual function (refer [BV04] for details). Since the cone of p.s.d matrices is self-dual, the dual program for an SDP is an SDP itself. The weak duality, like for other convex programs, follows from the construction of the Lagrangian. However, unlike LP, strong duality doesn't always hold for an SDP. The dual SDP needs to satisfy a set of conditions called *constraint qualifications*, for strong duality to hold. Slater's condition is one such widely used criteria, and we refer to the book [BV04] for these details.

SDP 2.2 (Primal)

$$\min \langle C, X \rangle$$

subject to

$$\langle A_i, X \rangle = b_i, \forall i \tag{2.12}$$

$$X \succeq 0\,. \tag{2.13}$$

SDP 2.3 (Dual)

$$\max \ \langle \mathbf{b}, \mathbf{y} \rangle$$

subject to

$$Y = C - \sum_i y_i A_i \tag{2.14}$$

$$Y \succeq 0\,. \tag{2.15}$$

The dual variables y_i correspond to the constraint in eq. (2.12) and Y corresponds to eq. (2.13). As discussed earlier, for combinatorial optimization problems, the SDP is typically written as a relaxation to an integer program with a specific intended solution. The intended solution (say $\mathbf{u}$) corresponds to a primal feasible solution $X = \mathbf{u}\mathbf{u}^T$. For many problems, we aim to show that the optimal primal solution is indeed the integral solution.

Fact 2.6 *The primal solution $X = \mathbf{u}\mathbf{u}^T$ is the unique solution to the SDP 2.2 if there exists a Y such that it satisfies constraints in SDP 2.3, with $\langle C, X \rangle = \mathbf{b}^T \mathbf{y}$, and has $\mathsf{rank}(Y) = n - 1$ (i.e. $\lambda_2(Y) > 0$).*

Proof: This is a folklore statement, and a proof of it can be found in Lemma 2.3 of [LV18]. □

2.4.3 Lasserre/Sum-of-squares(SOS) hierarchy

A yet another approach to relax QP 2.1 is by considering it as a system of polynomial inequalities and convexifying the set of feasible solutions by a set of probability distributions. The space of probability distribution is $\mathbb{R}^{2^n}$ and, therefore, intractable as such. A natural relaxation is to only look at space of distributions in $\mathbb{R}^{2^r}$ where r is a constant. This is what r^{th} level moments of this distribution let us do. Our moment approach for SDP in Section 2.4.1 can be thought as doing this where $Y_i = \mathbb{E}[x_i]$ and $Y_{ij} = \mathbb{E}[X_{ij}]$. The moment terms and the sum-of-squares constraint in eq. (2.11) can be generalized to r-tuples to give a similar SDP relaxation. This relaxation is called level-r Lasserre/Sum-of-squares (SOS) relaxation.

2.5 SDP/Lasserre hierarchies in approximation algorithms

Since the breakthrough result of Goemans and Williamson [GW95], SDP has been widely used in approximation algorithms, the works [FG95, KMS98, ARV04] etc. are a few notable ones.

SDP has also been the tool of choice for exact recovery in semi-random models. Starting from the fundamental works of exact recovery for the planted clique problem [FK00], for the planted bisection problem [FK01], for Stochastic Block Models [ABH16] etc., and many of the works mentioned in Section 1.2 are based on SDP relaxations. A natural way to analyze these SDP relaxations is by constructing an optimal dual solution to prove the integrality of the primal relaxation. This idea has been explored in the works of [FK01, CO07, BCC$^+$10, ABBS14, ABH16, LV18], to state a few. We note that the task of constructing an optimal dual solution is problem-specific, and there is no generic way of doing this.

The Lasserre/SoS hierarchy is a strengthened SDP relaxation for nonlinear $0-1$ programs attributed to the works of Shor [Sho87], Nesterov [Nes00], Jean B. Lasserre [Las01] and Parrilo [Par03]. We refer the reader to the survey by Thomas Rothvoß [Rot13] for a detailed discussion. The Lasserre/SoS hierarchy has been used in variety of works [Chl07, CS08, HSS15, HSSS16, HKP$^+$17, KS17b] etc to yield state of art algorithms.

Chapter 3

Odd Cycle Transversal Problem

In this chapter we will study the Odd Cycle Transversal problem or equivalently the largest induced bipartite subgraph problem in semi-random models.

Problem 3.1 *Given a graph $G = (V, E)$, the Odd Cycle Transversal (OCT) problem asks to find the smallest set $S \subseteq V$ such that S has a non-empty intersection with every odd cycle of the graph.*

Removing these set of vertices S will result in a bipartite graph, and hence this problem is equivalent to finding the largest induced bipartite graph. Given a graph $G = (V, E)$, the problem of finding the largest induced bipartite subgraph of G is well known to be NP-hard (shown in [Yan78]). The problem is also related to the balanced biclique problem, where the task is that of finding the largest induced balanced complete bipartite subgraph. This problem has a lot of practical application in computational biology [CC00], bioinformatics [Zha08] and VLSI design [AM99].

Models and results

We now present our semi-random model which we call *threshold semi-random model*. The model is inspired from the [FK01] model (Definition 1.3) and generalizes the planted solution model (Definition 1.1) and the sandwich model (Definition 1.2).

Definition 3.1 (Threshold Semi-random model) *Given n, k, d, p, our planted bipartite graph is constructed as follows,*

1. *Let V be a set of n vertices. Fix an arbitrary subset $S \subset V$ such that $|S| = k$.*
2. *Add edges arbitrarily in the graph induced by S such that the resulting graph is a connected d-regular bipartite graph. Let S_1, S_2 denote the bipartite components.*

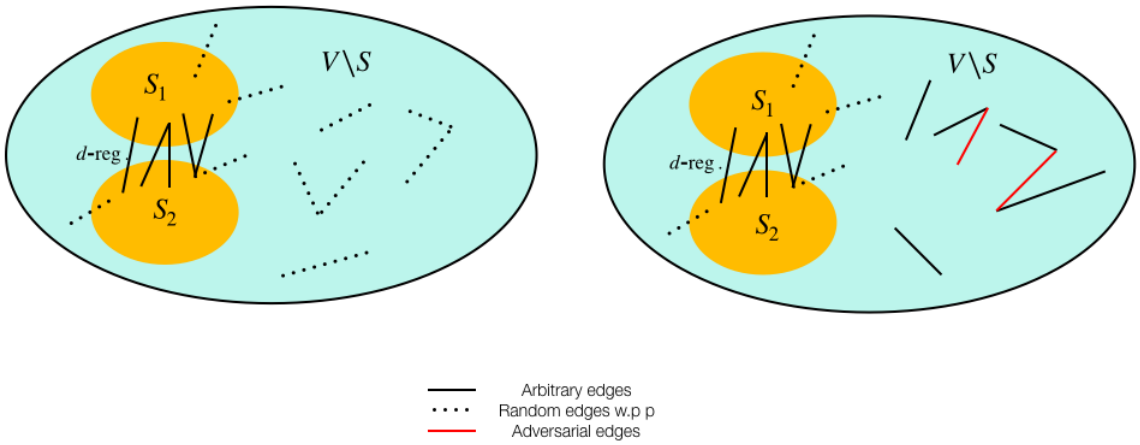

Figure 3.1: Planted solution model Definition 1.1 (left) and threshold semi-random model Definition 3.1 (right).

3. *For each pair of vertices in $S \times (V \setminus S)$, add an edge independently with probability p.*

4. *Add edges in $(V \setminus S) \times (V \setminus S)$ such that smallest eigenvalue of $\left(A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right)$ is greater than $-\left((1/2-\alpha)/(1/2+\alpha)\right)d$ where α is a small [1] positive constant (throughout this chapter we assume $\alpha \leq 1/6$).*

5. *Allow a monotone adversary to add edges in $(V \setminus S) \times (V \setminus S)$ arbitrarily.*

Remark 3.1 *Note that although it is not explicitly mentioned in the model construction, we will have $|S_1| = |S_2|$ since the graph induced on S is a d-regular bipartite graph.*

In [FK01] model, item 4 allows for any arbitrary graph in $(V \setminus S) \times (V \setminus S)$ (no assumptions on the graph induced on $V \setminus S$). However, with no further assumptions on $V \setminus S$, even for the special case of our problem i.e. planted independent set problem ($d = 0$), the best known algorithm in [FK01] model due to [MMT20] works only for $k = \Omega_p\left(n^{2/3}\right)$. However, our benchmark[2] is $k = \Omega\left(\sqrt{n}\right)$ and hence we look at a model with stronger assumptions than the [FK01] model.

A reasonable assumption to have is that the graph induced on $V \setminus S$ should not be close to having any induced bipartite subgraphs of degree at least d. Informally speaking, we want the graph induced on $V \setminus S$ to stand out from the planted graph on S. Our condition in item 4 can

[1] Note that the smaller the value of α, the weaker is this assumption.
[2] We refer to the proof overview for an explanation of setting this as a benchmark.

be interpreted as a way towards achieving this. If the smallest eigenvalue of the graph[3] induced on $V \setminus S$ is greater than $-d/2$, then the graph is indeed far from having an induced bipartite subgraph in $V \setminus S$ of smallest degree d. We can argue this by contradiction since otherwise, consider a vector with value 1 for one side of the alleged bipartition and -1 on the other side and 0 elsewhere achieves a Rayleigh Quotient of value $-d$. Using Fact 2.1, this implies that the smallest eigenvalue is $\leq -d$ contradicting that the smallest eigenvalue is at least $-d/2$.

Remark 3.2 *The threshold semi-random model in Definition 3.1 also captures the planted solution model in Definition 1.1; since in the case when $V \setminus S$ is chosen to be a $G_{(n-k),p}$ random graph, $\left(A_{(V\setminus S)\times(V\times S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right) = A_{(V\setminus S)\times(V\setminus S)} - \mathbb{E}\left[A_{(V\setminus S)\times(V\setminus S)}\right]$, and therefore the smallest eigenvalue of $\left(A_{(V\setminus S)\times(V\times S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right)$ is greater than $-2\sqrt{n}$ (as follows from Claim 2.1).*

We present our first result (details in Section 3.1) where we study the problem in planted solution model (Definition 1.1) and we give the result below.

Theorem 3.1 (Informal version of Theorem 3.3) *For a fixed k, p satisfying $k = \Omega_p\left(\sqrt{n}\right)$ and $p = \Omega\left(\sqrt{\log k/k}\right)$, there exists a deterministic algorithm that takes as input an instance generated as per planted solution model (Definition 1.1), and recovers the arbitrary planted set S exactly with high probability (over the randomness of the input) in time exponential in the number of small eigenvalues of the adjacency matrix (eigenvalues smaller than $-d/2 + 2\sqrt{n}$) of the graph induced on S.*

Remark 3.3 *For many special classes of instances such as, (i) when the probability $p = \Omega(1)$, (ii) when the planted graph is a complete bipartite graph like in the balanced biclique problem (iii) when the planted bipartite graph is random, or (iv) more generally when the planted graph is a d-regular expander graph; the number of these small eigenvalues is a constant and Theorem 3.1 allows efficient recovery (running time of the algorithm is polynomial in n).*

We will then proceed to present our main result (details in Section 3.2), which holds for both the planted solution model and the threshold semi-random model.

Theorem 3.2 (Informal version of Theorem 3.4) *For a fixed k, p sastisfying $k = \Omega_p\left(\sqrt{n \log n}\right)$, and $p = \Omega\left(\log k/k\right)^{1/6}$, there exists a deterministic algorithm that takes as input an instance generated by threshold semi-random model (Definition 3.1), and recovers the arbitrary planted set S exactly, in polynomial time and with high probability (over the randomness of the input).*

[3]Formally speaking, smallest eigenvalue of the adjacency matrix in item 4.

Related Works

The optimal long code test by Khot and Bansal [BK09] rules out any constant factor approximation for this problem. On the algorithmic front, casting the problem as a 2-CNF deletion problem, [AKRR90] gives a reduction to the min-multicut problem. This reduction gives us an $\mathcal{O}(\log n)$ approximation due to the work [GVY98], which was further improved to $\mathcal{O}\left(\sqrt{\log n}\right)$ in the work [ACMM05]. The work [GL21] gives an efficient randomized algorithm that computes an induced bipartite subgraph having $\left(1-\mathcal{O}\left(\sqrt{\varepsilon \log d}\right)\right)$ fraction of the vertices where d is the bound on the maximum degree of the graph. They also give a matching (up to constant factors) Unique Games hardness for certain regimes of parameters.

The problem is equivalent to finding the largest 2-colorable subgraph of a given graph and is also known as the partial 2-coloring problem. The work [GLR19] studies the problem in the Feige-Kilian semi-random model [GLR19], where a 2-colorable graph of size $(1-\varepsilon)n$ is planted. They give an algorithm that outputs a set $\mathcal{S}'$ such that $|\mathcal{S}'| \geq (1-\varepsilon c/p^2)n$ for $p = \Omega\left(\sqrt{\log n/n}\right)$ and $\varepsilon \leq p^2$ where c is a positive constant. Their algorithm is a partial recovery algorithm and works for the regimes when ε is small. We study the problem when $1-\varepsilon$ is small, and we give complete recovery for a large range of p. However, since our semi-random model makes stronger assumptions than the [FK01] model, we don't make any comparisons.

A related problem is that of finding the largest complete balanced bipartite subgraph called the balanced biclique problem, which we discuss next.

Balanced biclique problem: In the balanced complete bipartite subgraph problem (also called the balanced biclique problem), we are given a graph on n vertices and a parameter k, and the problem then asks whether there is a complete bipartite subgraph that is balanced with k vertices in each of the bipartite components. The problem was studied when the underlying graph is a bipartite graph, and shown to be NP-complete by a reduction from the CLIQUE problem in the works [GJ79, Joh87]. They additionally note that the balanced constraint is what makes the problem hard. If we remove the balanced constraint, the problem can be reduced to finding a maximum independent set in a bipartite graph. The latter problem admits a polynomial-time solution using the matching algorithm. The work [FK04] shows that this problem of finding a maximum balanced biclique is hard to approximate within a factor of $2^{(\log n)^{\delta}}$ for some $\delta > 0$, under the assumption that $\mathsf{3SAT} \notin \mathsf{DTIME}\left(2^{n^{3/4+\varepsilon}}\right)$ for some $\varepsilon > 0$. Recently, the work [Man17] showed that one cannot find a better approximation than $n^{1-\varepsilon}$, assuming the *Small Set Expansion Hypothesis* and that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ for every constant $\varepsilon > 0$.

A related problem is the maximum edge biclique problem, where we are asked to find whether

G contains a biclique with at least k edges. This problem was also shown to be NP-hard in the work [Pee03].

Given these intractability results for general graphs, there has been some success in special classes of graphs. In graphs with constant arboricity, the work [Epp94] gives a linear time algorithm that lists all maximal complete bipartite subgraphs. In a degree bounded graph, the work [TSS02] gives a combinatorial algorithm for the balanced biclique problem that runs in time $\mathcal{O}\left(n2^d\right)$. Another systematic approach, however, is to consider planted and semi-random models for the problem. In the work [Lev18], they study the planted version of the problem, which, they call "hidden biclique problem". Their model is similar to our model in Definition 1.1; however, we consider an arbitrary d-regular bipartite graph instead of a complete bipartite graph. They give a linear-time combinatorial algorithm that finds the planted hidden biclique with high probability (over the randomness of the input instance) for $k = \Omega\left(\sqrt{n}\right)$. Their algorithm builds on the "Low Degree Removal" algorithm, due to Feige and Ron [FR10] which finds planted clique in linear time.

Proof Overview

Benchmarks: We start by noting that the OCT problem is a generalization of the planted independent set and the planted balanced biclique problem. For $d = 0$, it reduces to recovering a planted independent set and hence we do not expect efficient algorithms for $k = o\left(\sqrt{n}\right)$ [FGR+13, BHK+16]. For $k = \Omega\left(\sqrt{n}\right)$, both the special cases of the problem, the planted independent set problem [AKS98, FK00], and the planted balanced biclique problem [Lev18] admit a polynomial-time recovery algorithm. So it is natural to consider $k = \Omega\left(\sqrt{n}\right)$ as a benchmark for recovery and look for algorithms in this regime.

The other interesting regime consideration comes by viewing this problem as a special case of the densest k-subgraph (DkS) problem. In DkS problem (in the planted model setting), the planted graph is an arbitrary graph that has average degree d and the algorithmic task is to recover this planted graph. When $d \gg pk$, the OCT problem can be viewed as the densest k-subgraph (DkS), and for $d \ll pk$, the OCT problem can be viewed as sparsest k-subgraph problem (studying the complement of this graph would be an instance of DkS problem). However, this general DkS problem is information-theoretically unsolvable for $d = pk$ [CX16]. Therefore we focus our attention on the case when $d \approx pk$. So the question we ask is whether we can use the additional[4] bipartite structure of the planted graph to recover the planted graph in $d \approx pk$ (including $d = pk$) regimes.

[4]additional w.r.t the DkS problem

Detecting planted bipartitions: We start by considering the detection version of the problem in the interesting regimes for this problem i.e. $k = \Omega(\sqrt{n})$ and $d \approx pk$. We note that the *detection problem* i.e. detecting the presence of bipartite graph as constructed in *planted solution model* Definition 1.1 against the null hypothesis of Erdős-Rényi graph $G_{n,p}$, is easy when $k = \Omega(\sqrt{n})$. Formally one notes that given two distributions

$$H_0 : G \sim G(n,p) \text{ against } H_1 : G \sim G(n,k,d,p) \text{ as per Definition 1.1,}$$

the test, which outputs H_1 when $\lambda_1(G) \leq -d$ and H_0 otherwise, is correct almost surely for $d \approx pk$ and $k \geq c\sqrt{n}/p$ where $c > 0$ is a large enough constant. This is because for a $G_{n,p}$ graph, the smallest eigenvalue is greater than $-2\sqrt{n}$ almost surely (from Claim 2.1), while for a graph with planted bipartite subgraph, the smallest eigenvalue is smaller than $-d$ since the vector $\mathbb{1}_{S_1} - \mathbb{1}_{S_2}$ already achieves Rayleigh Quotient of value $-d$ (using Fact 2.1).

Spectral Approaches: However, as expected, the exact recovery problem is more challenging. A natural spectral approach (where we use eigenvalues and eigenvectors), which has been used for planted models, e.g., [AKS98, McS01], relies on the fact that there is sufficient eigengap to apply results from perturbation theory (as discussed in Chapter 2). However, since the planted bipartite graph is arbitrary, there can be many eigenvalues close to $-d$ and hence there is no guarantee of a sufficient eigengap.

A possible approach then is to use all these eigenvectors with eigenvalues close to $-d$ to recover the planted set. To formalize this, we define the threshold rank of the graph [5]

Definition 3.2 *For $\tau \in [0,d]$, we define $\mathsf{rank}_{\leq -\tau}(G) = |\{i : \lambda_i(G) \leq -\tau\}|$.*

We let $P = \left\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(L_\tau)}\right\}$ (the bottom L_τ vectors) denote the set of eigenvectors of $A|_{S\times S}$ with eigenvalues smaller than the threshold τ where $L_\tau = \mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right)$. We call these vectors as *τ-threshold rank eigenvectors* of $A|_{S\times S}$.

Claim 3.1 *For the planted bipartite graph on set S, we have that $\mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right) \leq \dfrac{\gamma pk^2}{2\tau^2}$, where $d = \gamma pk$.*

Proof: It is a well known fact that the spectrum of bipartite graphs (in our case $A|_{S\times S}$) is symmetric around 0. Therefore the number of eigenvalues with absolute value greater then or equal to τ is given by $2\,\mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right)$. Using the fact that these eigenvalues are of a d-regular

[5] We mean adjacency matrix corresponding to the graph.

graph and hence lie in the interval $[-d, d]$, we have that,

$$2\tau^2 \mathsf{rank}_{\leq -\tau}(A|_{S\times S}) \leq \sum_i \lambda_i^2(A|_{S\times S}) \leq \left\| A|_{S\times S} \right\|_F^2 = kd\,.$$

Hence we have that,

$$\mathsf{rank}_{\leq -\tau}(A|_{S\times S}) \leq \frac{kd}{2\tau^2} = \frac{\gamma p k^2}{2\tau^2}$$

□

Now we use these *threshold rank eigenvectors* to recover the planted set by the *subspace enumeration* technique, which has been used in the works of [KT07, AG11, Kol11, KLT17]. Here we first identify that the vector $\mathbf{u} = \mathbb{1}_{S_1} - \mathbb{1}_{S_2}$ has a large projection on the space spanned by τ-threshold rank eigenvectors of A (for choice of $\tau = -d/2 + 2\sqrt{n}$). Note that this vector $\mathbf{u}$ identifies the planted set and therefore we call it the *signed indicator vector*. We then do a standard ε-net construction (refer [Ver18]) to find a vector $\mathbf{y}$ close to $\mathbf{u}$ and thus recover a large fraction of planted set S (Lemma 3.1). We can recover the remaining set of vertices by an argument due to the work [GLR19] where they distinguish vertices by the size of matching in induced neighborhoods (Section 3.1.2). Putting all this together, we prove our result in Theorem 3.3. We describe the details for the subspace enumeration technique in Section 3.1.

The running time of the subspace enumeration approach is exponential in L_τ. Now for a constant γ we have $L_\tau = \mathcal{O}(1/p)$ for $\tau = \Omega(pk)$ (follows from Claim 3.1). Therefore, for many special classes of instances such as, (i) when the probability $p = \Omega(1)$ and $\gamma = \Omega(1)$, (ii) when the planted graph is a complete bipartite graph (this is the balanced biclique problem) and $\gamma = \Omega(1)$, (iii) when the planted bipartite graph is random and $\gamma = \Omega(1)$ or (iv) more generally when the planted graph is a d-regular expander graph and $\gamma = \Omega(1)$; we have $L_\tau = \mathcal{O}(1)$ and this already gives us a polynomial-time algorithm. However, the case when $\gamma \leq 2/3$ is fairly simple and a combinatorial argument can work in those regimes (as we show in Section 3.2.2).

However, as we pointed earlier, we want to solve the problem in $p = o(1)$ regimes. Also, we want to solve the problem for arbitrary graphs and not just for a special class of graphs. To accomplish this, we shift our focus to the SDP relaxations.

SDP Relaxation: We consider the following SDP 3.1 relaxation. We construct its dual SDP 3.2 (refer to Section 3.3.1 for more details on this dual construction).

SDP 3.1 (Primal)

$$\min \sum_{\{i,j\}\in E} 2\langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

subject to

$$\sum_{i\in V} \|\mathbf{x}_i\|^2 = 1 \tag{3.1}$$

$$\langle \mathbf{x}_i, \mathbf{x}_j\rangle \leq 0 \qquad \forall \{i,j\} \in E\,. \tag{3.2}$$

SDP 3.2 (Dual)

$$\max\ \beta$$

subject to

$$Y = A - \beta I + \sum_{\{i,j\}\in E} B_{ij}\left(\mathbb{1}_{ij} + \mathbb{1}_{ji}\right) \tag{3.3}$$

$$B_{ij} \geq 0, \qquad \forall \{i,j\} \in E \tag{3.4}$$

$$Y \succeq 0\,. \tag{3.5}$$

We denote by X the primal SDP matrix. Let $\mathbf{x}_i$ denote the vector corresponding to vertex i such that $X_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j\rangle$. We consider this feasible integral solution (denoted by $X = \mathbf{g}\mathbf{g}^T$, where $\mathbf{g} \in \mathbb{R}^n$ s.t $g_i = 1/\sqrt{k}$ for $i \in S_1$, $g_i = -1/\sqrt{k}$ for $i \in S_2$ and 0 otherwise) to the SDP by setting,

$$\mathbf{x}_i = \begin{cases} \hat{e}/\sqrt{k} & \text{if } i \in S_1 \\ -\hat{e}/\sqrt{k} & \text{if } i \in S_2 \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\hat{e}$ is some unit vector. In SDP 3.2, the Lagrange multipliers β and B_{ij}'s are our dual variables and Y is the dual SDP matrix. By $\mathbb{1}_{ij}$ we mean an indicator matrix which is 1 for (i,j) entry and 0 elsewhere. For clarity, we will denote $\sum_{\{i,j\}\in E} B_{ij}\left(\mathbb{1}_{ij} + \mathbb{1}_{ji}\right)$ by a matrix B. Let $\mathsf{SDPOPT}\,(G)$ denote the optimal value of the primal SDP. From the proposed integral solution we know that $\mathsf{SDPOPT}\,(G) \leq -d$. For any feasible solution to the dual SDP 3.2, by weak duality, we know that $\beta \leq \mathsf{SDPOPT}\,(G) \leq -d$.

Constructing good dual SDP solutions to certify optimality of Primal: Broadly speaking, a popular technique in analyzing SDP relaxations (like SDP 3.1) is to show optimality by constructing a dual solution that matches the $\mathsf{SDPOPT}\,(G)$ value of the primal in a manner that the dual matrix Y has rank $n-1$, (see Fact 2.6). This framework already imposes a list of conditions which can be used to characterize the dual variables

1. $\beta = -d$
2. $\left\langle \mathbf{g}\mathbf{g}^T, Y\right\rangle = 0$
3. $Y \succeq 0$
4. $\lambda_2\left(Y\right) > 0\,.$

To meet condition (1), we set $\beta = -d$ to match the primal objective value of $\mathsf{SDPOPT} = -d$

(strong duality objective condition). We expand upon complementary slackness condition (2) as,

$$\left\langle \mathbf{g}\mathbf{g}^T, Y \right\rangle = \mathbf{g}^T Y \mathbf{g} = \mathbf{g}^T \left(A + dI\right) \mathbf{g} + \mathbf{g}^T B \mathbf{g} = 0 + \mathbf{g}^T B \mathbf{g} = \mathbf{g}^T B \mathbf{g} \qquad \text{(Using } \beta = -d\text{)} \ .$$

Therefore, from condition (2), we have

$$\sum_{i \in S} \sum_{\substack{j \in S \\ \{i,j\} \in E}} B_{ij} = 0 \tag{3.7}$$

and since $B_{ij} \geq 0$ this implies that $B_{ij} = 0$ for all $(i, j) \in E(S_1, S_2)$.

For certain problems in semi-random models such as the planted clique problem [FK00], community detection in SBM [ABH16], these list of conditions above suffices to handcraft a feasible dual solution. For these problems, one can also show that the handcrafted dual solution satisfies conditions equivalent to (3) and (4), by typically using some standard results for random matrix bounds and concentration inequalities. In our setup, imposing condition (3) requires

$$\lambda_{\min}\left(Y\right) \geq \lambda_{\min}\left(A\right) + d + \lambda_{\min}\left(B\right) \ . \tag{3.8}$$

However, if $\lambda_{\min}(A) \leq -d - 2\sqrt{pn}$ (which is possible[6]), condition (3) may not hold (as per choice of B_{ij}'s dictated from eq. (3.7)).

Thus, satisfying Constraint (3.8) seems to require more work. Motivated by success stories in other recovery problems like the planted bisection problem [FK01], coloring semi-random graphs [CO07], decoding binary node labels from censored edge measurements [ABBS14], and planted sparse vertex cuts [LV18], one approach may be to construct some sort of dual matrix by giving "meaning" to the dual variables. Such an approach would work if the planted bipartite graph were also random. Since random graphs are good expanders, there would only be a single eigenvector that disobeys eq. (3.8), and we can choose the dual variables constructively to handle this.

However, for arbitrary graphs, to handle the situation discussed above, we need a more principled approach to deal with the eigenvectors of the planted graph having eigenvalue close to $-d$ (note that there can be many such eigenvectors). This is our primary motivation for defining threshold rank and *threshold rank eigenvector* as in Definition 3.2.

In Section 3.2 we present our "principled" approach in dealing with the *threshold rank*

[6]The smallest eigenvalue from A can be $-d$ and from the rest of graph $-2\sqrt{n}$. Now. using Weyl's inequality the only guarantee we have is that the eigenvalue of A is atleast $-d - 2\sqrt{n}$

eigenvectors close to $-d$ such that the strong duality still holds.. Our proofs use the spectral properties of bipartite graphs and random graphs to show the existence of an optimal dual solution having rank $n-1$.

3.1 Exact Recovery using Subspace Enumeration

In this section, we study the problem of exact recovery of a planted bipartite graph in the *planted solution* model constructed according to Definition 1.1. We focus on the regimes when the planted bipartite graph has degree $d = \gamma pk$ and when the planted set has size $k = \Omega_p(\sqrt{n})$. We let L_τ be the threshold rank of $A|_{S\times S}$ as defined in Definition 3.2 for some choice of threshold τ. In Section 5.1, we give a procedure to recover $(1-\delta)$ fraction of vertices in S for any given value of $\delta > 0$. Using arguments similar to [GLR19], we can recover the remaining set of vertices (Lemma 3.5).

Theorem 3.3 *For a fixed k, p where $k \geq \dfrac{256\sqrt{n}}{\gamma^2 p^3}$ and $(k/\log k) \geq 25/p^2$, and choice of $\tau = -d/2 + 2\sqrt{n}$, there exists a deterministic algorithm which can recover the planted set S in an instance generated as per planted solution model (Definition 1.1), exactly with high probability (over the randomness of the input) in time $\mathcal{O}\left(\mathsf{poly}(n)k^{(L_\tau+1)}\right)$.*

We note that the constants in Theorem 3.3 have not been optimized for, and these specific values are a result of choices we make for ease of calculation.

3.1.1 Partial recovery of the planted set

In this setting, the vector $\mathbf{u} = \mathbb{1}_{S_1} - \mathbb{1}_{S_2}$, which also indicates the planted set has small Rayleigh quotient (of value $-d$). We recall that we had referred to this vector $\mathbf{u}$ as the *signed indicator vector* for our planted set S. Although $\mathbf{u}$ is not an eigenvector for the entire matrix A, we can still show that it has a large projection on the subspace formed by the bottom $L_{\tau'} = L_\tau + 1$ eigenvectors[7] (having eigenvalues smaller than τ'). Therefore we can do a brute force search in this space via the subspace enumeration technique along the lines of [KT07, AG11, Kol11, KLT17] and attempt to recover a vector that is close to this signed indicator(distance to the $\mathbf{u}$ is small, see Lemma 3.3) for the planted set. We then use this vector to recover $(1-\delta)$ fraction of the planted set S for any given value of $\delta > 0$ as,

[7]The term L_τ was earlier defined only for $A|_{S\times S}$ but here we use it for the entire A also with the subscript τ for $A|_{S\times S}$ and τ' for A respectively. Therefore it should be clear from the context which one we are talking about.

Lemma 3.1 *For an instance of planted graph given by the planted solution model (Definition 1.1) and a parameter $\delta > 0$ in regimes where $k \geq ((8-2\delta)/\delta\gamma p)\, 2\sqrt{n}$, there exists a deterministic algorithm which can recover at least $(1-\delta)$ fraction of the planted vertices in S.*

Lemma 3.2 *For $\tau' = \tau - 2\sqrt{n}$ we have,* $\mathsf{rank}_{\leq -\tau'}(A) \leq \mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right) + 1$

Proof: We recall that we let τ to be the threshold for the matrix $A|_{S\times S}$ such that $L_\tau = \mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right)$. Note that this is also equal to $\mathsf{rank}_{\leq -\tau}(A_{S\times S})$. From here, we relate to $\mathsf{rank}_{\leq -\tau}(A)$ in a series of steps where we recall that,

$$A = A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T + p\mathbb{1}\mathbb{1}^T + R$$

The $-p\mathbb{1}_S\mathbb{1}_S^T$ term only affects the $\mathbb{1}_S$ eigenvector of $A_{S\times S}$ such that the corresponding eigenvalue is shifted form d to $d - pk$. Therefore, we have that for the same value of τ as for $A_{S\times S}$, $\mathsf{rank}_{\leq -\tau}\left(A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T\right) \leq L_\tau + 1$, to accommodate for the $\mathbb{1}_S$ eigenvector. Next we consider the perturbation matrix R then we have that for choice of $\tau' = \tau - 2\sqrt{n}$ we have that

$$\mathsf{rank}_{\leq -\tau'}\left(A|_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T + R\right) \leq \mathsf{rank}_{\leq -\tau}\left(A|_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T\right).$$

The choice of τ and τ' are such that we account for the shift in eigenvalues due to R since we know that almost surely $\|R\|_2 \leq 2\sqrt{n}$ (from Claim 2.1). Finally we account for the term $p\mathbb{1}\mathbb{1}^T$ by using Weyl's inequality (Fact 2.4) where $B = p\mathbb{1}\mathbb{1}^T$ and $C = A|_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T$, here $\lambda_1(B) = 0$, and we get that,

$$\lambda_1(C) \leq \lambda_1(C+B) \leq \lambda_2(C) \leq \ldots \leq \lambda_{n-1}(C+B) \leq \lambda_n(C) \leq \lambda_n(C+B).$$

Therefore we have that,

$$\mathsf{rank}_{\leq -\tau'}(A) \leq \mathsf{rank}_{\leq -\tau'}\left(A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T + R\right).$$

Putting everything together we obtain that, $\mathsf{rank}_{\leq -\tau'}(A) \leq \mathsf{rank}_{\leq -\tau}\left(A|_{S\times S}\right) + 1$ □

Lemma 3.3 *Let $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ denote the eigenvectors of A, then there exists a vector $\mathbf{y}' \in \mathsf{span}\left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(L_{\tau'})}\right\}$ such that for a positive constant $\rho > 2\sqrt{n}$ and choice of $\tau' = -d/2$,*

$$\|\mathbf{y}' - \mathbf{u}\|^2 \leq \frac{\rho k}{d/2 + \rho}.$$

Proof: We express the *signed indicator vector* $\mathbf{u}$ in the basis of these eigenvectors (of unit norm) as,

$$\mathbf{u} = c_1 \mathbf{v}^{(1)} + \dots c_n \mathbf{v}^{(n)} \,. \tag{3.9}$$

for some constants $c_1, \dots, c_n$. Consider the matrix $A' = A + (d+\rho)\, I$ where ρ is a positive constant larger than $2\sqrt{n}$.[8] Since the identity matrix only shifts the eigenvalues we have that,

$$\begin{aligned} \rho = \frac{\mathbf{u}^T A' \mathbf{u}}{\mathbf{u}^T \mathbf{u}} &= \frac{\left(\sum_i c_i \mathbf{v}^{(i)}\right) A' \left(\sum_i c_i \mathbf{v}^{(i)}\right)}{\|\mathbf{u}\|^2} = \frac{\sum_{i=1}^n \left(\lambda_i (A) + d + \rho\right) c_i^2 \left\|\mathbf{v}^{(i)}\right\|^2}{k} \\ &\geq \frac{\left(\lambda_{L_{\tau'}+1}(A) + d + \rho\right) \sum_{i=L_{\tau'}+1}^n {c_i}^2}{k} \,. \end{aligned} \tag{3.10}$$

The last inequality uses the fact that the eigenvalues are non-negative; therefore, at this point, we need that $\rho \geq 2\sqrt{n}$.

Consider the space spanned by the first $L_{\tau'}$ eigenvectors as $\mathsf{span}\left\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(L_{\tau'})}\right\}$. If we consider the vector $\mathbf{y}' = c_1 \mathbf{v}^{(1)} + \dots c_{L_{\tau'}} \mathbf{v}^{(L_{\tau'})}$ (for the same constants $c_1, \dots, c_{L_{\tau'}}$ as in eq. (3.9)), by construction it lies in the space spanned by these bottom $L_{\tau'}$ eigenvectors of A i.e.,

$$\mathbf{y}' = c_1 \mathbf{v}^{(1)} + \dots c_{L_{\tau'}} \mathbf{v}^{(L_{\tau'})} \quad \text{belongs to} \quad \mathsf{span}\left\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(L_{\tau'})}\right\} \,.$$

Now if we consider the distance between these vectors we get,

$$\|\mathbf{y}' - \mathbf{u}\|^2 = \left\| \sum_{i=L_{\tau'}+1}^{n} c_i \mathbf{v}_i \right\|^2 = \sum_{i=L_{\tau'}+1}^{n} c_i^2 \,. \tag{3.11}$$

Using our choice of $\tau' = -d/2$ in eq. (3.10) and eq. (3.11) we have that,

$$\|\mathbf{y}' - \mathbf{u}\|^2 = \sum_{i=L_{\tau'}+1}^{n} c_i^2 \leq \frac{\rho k}{\left(\lambda_{L_{\tau'}+1}(A) + d + \rho\right)} \leq \frac{\rho k}{d/2 + \rho} \,.$$

□

Hence there exists a vector $\mathbf{y}' \in \mathbb{R}^{L_{\tau'}}$, which is close to a vector that indicates the set S. Next in Lemma 3.4, we show how to find such a $\mathbf{y}'$, whose existence we have argued in Lemma 3.3. We do so by a brute force search over the space spanned by these $L_{\tau'}$ eigenvectors. We cannot search over the infinite points in the space as such, but we can construct an ε-net

[8] We shift the matrix so that eigenvalues of A' are ≥ 0.

and choose a value of ε such that we get a point in this space for which the distance to $\mathbf{y}'$ is smaller than ε.

Lemma 3.4 *There exists a deterministic algorithm running in time* $\mathcal{O}\left(k^{L'_\tau}\right)$ *which finds a vector* $\mathbf{y} \in \mathsf{span}\left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(L_{\tau'})}\right\}$ *such that for a value of* $\rho > 2\sqrt{n}$,

$$\|\mathbf{y} - \mathbf{u}\|^2 \leq \frac{2\rho k}{d/2 + \rho}\,.$$

Proof: We build an ε-net such that for any $\mathbf{v} \in \mathsf{span}\left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(L_{\tau'})}\right\}$ we have another $\mathbf{v}'$ which belongs to the ε-net and is also close to $\mathbf{v}$ such that,

$$\|\mathbf{v} - \mathbf{v}'\| \leq \varepsilon\,.$$

For unit norm vectors, the bound on number of points in an ε-net is given by $(3/\varepsilon)^{L_{\tau'}}$(Corollary 4.2.13, [Ver18]). Since our vector $\mathbf{u}$ has squared norm k, we consider ball of radius k and the volume is scaled by factor of $k^{L\tau'}$.

Since the space is $L_{\tau'}$-dimensional the number of points in the space for such an ε-net is upper bounded by $(3k/\varepsilon)^{L_{\tau'}}$(Corollary 4.2.13, [Ver18]). Thus for the vector $\mathbf{y}'$ in Lemma 3.3 we can find a vector $\mathbf{y}$ such that,

$$\|\mathbf{y} - \mathbf{u}\|^2 = \|(\mathbf{y} - \mathbf{y}') + (\mathbf{y} - \mathbf{u})\|^2 \leq \|\mathbf{y} - \mathbf{y}'\|^2 + \|\mathbf{y}' - \mathbf{u}\|^2 \leq \varepsilon^2 + \frac{\rho k}{d/2 + \rho}\,.$$

We choose $\varepsilon^2 = \rho k/\left(d/2 + \rho\right)$, and hence $\varepsilon = \sqrt{\rho k/\left(d/2 + \rho\right)}$. Since $\rho > 2\sqrt{n}$ and $d \leq k$ the number of points (denoted by $\mathcal{N}$) are bounded by,

$$\begin{aligned}\mathcal{N} &\leq \left(\frac{3k}{\varepsilon}\right)^{L'_\tau} = \left(3\sqrt{k}\right)^{L'_\tau}\left(\frac{d/2+\rho}{\rho}\right)^{L'_\tau/2} \leq \left(3\sqrt{k}\right)^{L'_\tau}\left(1+\frac{d}{2\rho}\right)^{L'_\tau/2} \leq \left(3\sqrt{k}\right)^{L'_\tau}\left(1+\frac{k}{4\sqrt{n}}\right)^{L'_\tau} \\ &\leq \left(3\sqrt{k}\right)^{L'_\tau}\left(1+\frac{\sqrt{n}}{4}\right)^{L'_\tau/2} \leq \left(3\sqrt{k}\right)^{L'_\tau}\left(\frac{k}{2}\right)^{L'_\tau/2} \leq (2k)^{L'_\tau} \qquad \left(\text{Using } \sqrt{n} \leq k \leq n\right)\,.\end{aligned}$$

Therefore the number of points are $\mathcal{O}\left(k^{L_{\tau'}}\right)$. Hence we can construct this ε-net in time $\mathcal{O}\left(k^{L_{\tau'}}\right)$. □

Proof: [Proof of Lemma 3.1] We wish the distance between $\mathbf{y}$ and $\mathbf{u}$ to be smaller than $\delta k/2$ for any arbitrary choice of $\delta > 0$.

$$\frac{2\rho k}{d/2+\rho} = \frac{4\rho k}{\gamma p k + 2\rho} \leq \frac{\delta k}{2} \text{ which holds if we choose } \rho \text{ as } \rho \leq \frac{\delta\gamma p k}{8 - 2\delta}\,. \tag{3.12}$$

From Lemma 3.3 we know that $\rho \geq 2\sqrt{n}$, which holds if,

$$k \geq \left(\frac{8-2\delta}{\delta\gamma p}\right) 2\sqrt{n}\,.$$

Next we formalize that this vector $\mathbf{y}$ closely indicates our planted set S. We sort the vector $\mathbf{y}$ by absolute value and pick the top k entries in a set $\mathcal{S}'$. let t be a threshold such that we have $\mathcal{S}' = \{i : |y_i| \geq t\}$.

We denote B as the bad set of vertices, $B \stackrel{\text{def}}{=} S \setminus \mathcal{S}'$. We note that $|S \setminus \mathcal{S}'| = |B| = |\mathcal{S}' \setminus S|$ since both S and $\mathcal{S}'$ have cardinality k. We let η be the fraction of these bad vertices which belong to S_1 and $(1-\eta)$ fraction then belong to S_2. Therefore,

$$\|\mathbf{y} - \mathbf{u}\|^2 = \sum_{i \notin S} {y_i}^2 + \sum_{i \in S_1} (y_i - 1)^2 + \sum_{i \in S_2} (y_i + 1)^2 \ .$$

Doing a term by term analysis we get that,

$$\begin{aligned}
\sum_{i \notin S} y_i^2 &\geq \sum_{i \in \mathcal{S}', i \notin S} y_i^2 \geq |B|\, t^2 \\
\sum_{i \in S_1} (y_i - 1)^2 &\geq \sum_{i \in S_1, i \notin \mathcal{S}'} (y_i - 1)^2 \geq \eta\, |B| \min\left\{(t-1)^2, (t+1)^2\right\} \geq \eta\, |B|\, (1-t)^2 \\
\sum_{i \in S_2} (y_i + 1)^2 &\geq \sum_{i \in S_2, i \notin \mathcal{S}'} (y_i + 1)^2 \geq (1-\eta)\, |B| \min\left\{(1-t)^2, (1+t)^2\right\} \geq (1-\eta)\, |B|\, (1-t)^2 \,.
\end{aligned}$$

where the first inequality holds because $u_i = 0$ and $y_i^2 \geq t^2$ and in the second and third inequality we use $u_i = -1$ and $u_i = 1$ respectively and $y_i \in [-t, t]$. The lower bound in these inequalities hold for $t \leq 1$; which is true, as can be seen form the fact that the algorithm picks top k entries of $\mathbf{y}$, and $t > 1$ gives a contradiction to the fact that vector $\mathbf{u}$ (which lies inside our ε-net of radius $\sqrt{k}$) has squared norm $\leq k$. Therefore we get that,

$$\frac{2\rho k}{d/2 + \rho} \geq \|\mathbf{y} - \mathbf{u}\|^2 \geq |B| \left(t^2 + (1-t)^2\right) \geq \frac{|B|}{2} \,.$$

The last inequality holds by observing that $t = 1/2$ minimizes that expression, and thus we have that the set of bad vertices $|B| \leq \delta k$. □

3.1.2 Algorithm for full recovery

In Lemma 3.1 we output a set $\mathcal{S}'$ such that $|\mathcal{S}'| = k$ and $|\mathcal{S}' \cap S| \geq (1-\delta)\, k$ for any constant $\delta > 0$. In this section, we propose an algorithm that allows us to recover the whole of the

planted set (in appropriate parameter regimes). The main idea here is to distinguish between vertices that belong to S and those which don't by considering the size of a maximum matching in subgraph induced on the neighborhood of vertex and the set $\mathcal{S}'$. This idea is used in the work [GLR19] in a similar vein to recover the vertices in a semi-random model for the same problem.

However, since $\mathcal{S}'$ is a function of the randomness of the input and not a fixed set, we cannot use the randomness of the input instance in our arguments. Therefore, we bound the size of matching to the fixed set S (where we are allowed to use the randomness of the instance) and factor in the worst-case contribution from $S \setminus \mathcal{S}'$.

Lemma 3.5 *Given a set $\mathcal{S}'$ of size k such that $S \cap \mathcal{S}' \geq (1-\delta)\,k$ for any $\delta > 0$ in the regimes of p, k discussed in Theorem 3.3, there exists a polynomial time deterministic algorithm that recovers the planted set S completely.*

Proof: Let $\mathsf{MM}\,(v, S)$ denote the maximum matching in the graph induced on $N(v) \cap S$. For a vertex $v \in V \setminus S$, the expected size of this maximum matching is given as,

$$\mathbb{E}\left[\mathsf{MM}\,(v, S)\right] \geq \frac{p^2 k}{2}\,.$$

This is because a d-regular bipartite graph has at least one perfect matching. We fix an arbitrary such matching. Now each of these matching edges are present in this neighborhood graph with probability p^2. The edges in the matching are independent of each other and therefore using the Chernoff bounds (Fact 2.3) we obtain,

$$\mathbb{P}\left[v \in V \setminus S : \mathsf{MM}\,(v, S) \leq \frac{p^2 k}{4}\right] \leq \exp\left(-\frac{p^2 k}{8}\right)\,.$$

This holds for a fixed vertex $v \in V \setminus S$; to make this claim for an arbitrary vertex in $V \setminus S$ we do a union bound to obtain,

$$\mathbb{P}\left[\exists v \in V \setminus S, \mathsf{MM}\,(v, S) \leq \frac{p^2 k}{4}\right] \leq k^2 \exp\left(-\frac{p^2 k}{8}\right)\,.$$

Hence for $p \geq 5\sqrt{(\log k)\,/k}$ we have a lower bound on the size of matching in the graph induced on the neighborhood of all $v \in V \setminus S$ and the set S, with high probability (over the randomness of the input). However as we mention earlier, we are interested in size of matching for a vertex $v \in V \setminus S$ and $\mathcal{S}'$. Since $|S \setminus S'| \leq \delta k$, and the vertices in matching edges need to be distinct the drop in the size of matching is at most δk. Therefore the size of matching for vertex $v \in V \setminus S$

and set $\mathcal{S}'$ can be lower bounded as,

$$\mathsf{MM}\left(v, \mathcal{S}'\right) \geq \left(\frac{p^2}{4} - \delta\right) k\,.$$

Now for a vertex $v \in S$ the size of matching in $N(v) \cap S$ is 0. This is because a matching edge in $N(v) \cap S$ along with the vertex v gives a triangle in S. This contradicts that S is a bipartite graph and hence it has no triangles. Therefore in the worst case, using the same argument as above, we can upper bound the size of maximum matching for vertex $v \in S$ in the graph $N(v) \cap \mathcal{S}'$ is,

$$\mathsf{MM}\left(v, \mathcal{S}'\right) \leq \delta k\,.$$

Therefore we can distinguish the set to which a vertex belongs if,

$$\frac{\gamma p^2}{4} - \delta > \delta, \text{ for convenience we take this gap to be } 2\delta \text{ then } \delta \leq \frac{\gamma p^2}{16}\,. \tag{3.13}$$

To satisfy eq. (3.12) we use the value of δ and from the condition eq. (3.13) above we get that for full recovery it is sufficient if $k \geq 256\sqrt{n}/\gamma^2 p^3$. □

Proof: [Proof of Theorem 3.3] Lemma 3.1 allows us to recover $(1-\delta)$ fraction of vertices in the planted set. Using Lemma 3.5 we can recover the remaining set of vertices. Hence for a fixed p, k where $p \geq 5\sqrt{(\log k)/k}$, $k \geq 256\sqrt{n}/\gamma^2 p^3$ we can recover the planted set exactly. The vector $\mathbf{y}$ used for recovery was constructed in Lemma 3.4 and this takes time exponential in $L'_\tau \log k$. Computing eigenvectors takes $\mathsf{poly}\,(n)$ time Therefore, the overall running time is $\mathcal{O}\left(\mathsf{poly}(n) k^{(L_\tau+1)}\right)$. □

3.2 Exact recovery in polynomial time using SDP

In this section, we consider the problem of recovering the planted bipartite graph constructed as per model Definition 3.1. The problem becomes non-trivial in $d \approx pk$ regimes, where the exact recovery problem in a more general setting of densest k-subgraph problem is information-theoretically unsolvable. Formally, this follows from Theorem 2.1, [CX16] by setting $d = qk = pk$ and setting $r = 1$ where q is the edge probability within the vertices of planted subgraph and a p is the edge probability when at least one of the vertex does not belong to the planted subgraph, and r is the number of clusters. However, in our problem, we use the specifics of the bipartite structure in hand to write an SDP relaxation (SDP 3.1) and show that we can use it to recover the planted set exactly. Formally we prove the following.

Theorem 3.4 *For a fixed p, k satisfying $k \geq \frac{256}{\alpha p^{7/2}}\sqrt{n \log n}$ and $p \geq 5\left((\log k) / (1/2+\alpha)^4 k\right)^{1/6}$, there exists a deterministic algorithm which recovers the planted set S in an instance generated as per threshold semi-random model (Definition 3.1), exactly with high probability (over the randomness of the input).*

We did not make any attempt to optimize the constants above (in Theorem 3.4), and the specific values we use are a result of choices we make for ease of calculation.

We break up our study of the problem into two different regimes based on the degree $d = \gamma pk$ of the planted graph, namely the low degree regime and high degree regimes.

3.2.1 High degree regimes

The main workhorse of our algorithm in high degree regimes is our SDP 3.1. In this section, we show how to construct an optimal dual to the SDP. By high degree regimes we mean that $\gamma \geq 1/2 + \alpha$. We recall that α is a small positive constant smaller than $1/6$.

3.2.1.1 Constructing an optimal dual

Our core idea is to to extend (by padding with 0's) the threshold rank eigenvectors of $A|_{S\times S}$ to be the eigenvectors [9] of the dual matrix Y. We recall the list of conditions (1)-(4) that dual variables had to follow as,.

1. $\beta = -d$
2. $\left\langle \mathbf{g}\mathbf{g}^T, Y \right\rangle = 0$
3. $Y \succeq 0$
4. $\lambda_2(Y) > 0$.

Now since the eigenvalues of $A|_{S\times S}$ lie in the interval $[-d, d]$, these threshold rank eigenvectors which are now also the *threshold rank eigenvectors* of Y with eigenvalue $d + \lambda_l$ have a non-negative quadratic form[10]. This way we attempt to satisfy feasibility and optimality condition (3) and (4) respectively.

For the remaining eigenvectors (other than threshold rank eigenvectors), we show (in Lemma 3.6) that if the matrix of non-negative dual variables B satisfies $\|B\|_2 = \tilde{\mathcal{O}}(pk)$[11], the quadratic form is indeed non-negative. The reasons for this condition will be evident in the proof of Lemma 3.6. This imposes another condition that we use to characterize our dual variables.

Therefore to show the feasibility of Y, it suffices to construct a dual solution B where

[9] With slight abuse of notation we will denote these padded vectors of length n also as $\mathbf{v}^{(l)}$.
[10] The quadratic form of vector $\mathbf{x}$ with a matrix Y is a number given by $\mathbf{x}^T Y \mathbf{x}$
[11] $\tilde{\mathcal{O}}(.)$ hides logarithmic factors.

$\|B\|_2 = \tilde{\mathcal{O}}(pk)$ and conditions as,

$$\sum_{i \in S} \mathbf{v}_i^{(l)} (A_{ij} + B_{ij}) = 0, \quad \forall j \in V \setminus S\,. \tag{3.14}$$

To aid further discussion, it is useful to note that eq. (3.14) considers

- L_τ different system of equations $\mathcal{E}_j$, one for each $\mathbf{v}^{(l)}$.
- Each system $\mathcal{E}_j$ involves $|S| \times |V \setminus S|$ variables B_{ij} where $i \in S$ and $j \in V \setminus S$.

Constructing an explicit dual solution doesn't seem easy for all these constraints. Therefore we try to show the existence of a feasible dual solution satisfying condition (1)-(4), eq. (3.14) and $\|B\|_2 = \tilde{\mathcal{O}}(pk)$.

We defer proving the existence of such B_{ij}'s to Section 3.2.1.2. However since the eq. (3.14) only concerns $B_{ij} \in S \times (V \setminus S)$. Therefore, we can set $B_{ij} = 0, \forall \{i,j\} \in (V \setminus S \times V \setminus S)$ for the purpose of satisfying eq. (3.14). Since eq. (3.7) already forces us to set $B_{ij} \in S \times S$ to be 0 we have,

$$B = B_{S\times S} + B_{S\times(V\setminus S)} + B_{(V\setminus S)\times S} + B_{(V\setminus S)\times(V\setminus S)} = B_{S\times(V\setminus S)} + B_{(V\setminus S)\times S}\,.$$

We will next show that under the assumption about existence of such B_{ij}'s, how we can proceed towards satisfying the optimality conditions for dual.

Fact 3.1 *Given T to be some set of orthonormal eigenvectors of a symmetric matrix M labeled as*

$$T = \left\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(n)}\right\},$$

to show that $M \succeq 0$ it is sufficient to show that $\mathbf{u}^T M \mathbf{u} \geq 0, \forall \mathbf{u} \in T$.

Lemma 3.6 *For p, k satisfying $k \geq \left(256/\alpha p^{7/2}\right)\sqrt{n \log n}$ and $p \geq 5\left((\log k)/\gamma^4 k\right)^{1/6}$, when $\lambda_{\min}\left(A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right) \geq -\tau$, $\gamma \geq 1/2+\alpha$ and for choice of $\tau = (d\,(1/2-\alpha)\,/\,(1/2+\alpha))$ and if there exists a B satisfying $\|B\| \leq 512\sqrt{n\,(\log k)}/p^{5/2}$, the dual matrix $Y \succeq 0$. Additionally, if the graph is connected, we have that $\lambda_2(Y) > 0$.*

Proof: We will proceed using Fact 3.1 and show that $\mathbf{u}^T Y \mathbf{u} \geq 0$, for all eigenvectors $\mathbf{u} \in T$ of the dual matrix Y with high probability (over the randomness of the input). We start with the *τ-threshold rank eigenvectors* of $A|_{S\times S}$ which are now also eigenvectors of Y (after padding them

with 0's). These are already orthonormal since $A_{S\times S}$ is symmetric. Also these eigenvectors have eigenvalue $d+\lambda_l \geq 0$ and therefore,

$$\mathbf{v}^{(l)^T} Y \mathbf{v}^{(l)} = \left\langle \mathbf{v}^{(l)}, Y\mathbf{v}^{(l)} \right\rangle = (d+\lambda_l) \left\| \mathbf{v}^{(l)} \right\|^2 \geq 0, \quad \forall\, \mathbf{v}^{(l)} \in P\,.$$

Now since the dual matrix Y is symmetric, we can extend the set of vectors P to a complete eigenbasis T for Y as follows. We include an eigenvector $\mathbf{x}$ in $T \setminus P$ if $\mathbf{x}\left\langle \mathbf{x}, \mathbf{v}^{(l)} \right\rangle = 0, \forall\, \mathbf{v}^{(l)} \in P$. Since, the eigenvectors added will be orthogonal to zero-padded extended vectors, this also implies that $\left\langle \mathbf{x}_S, \mathbf{v}^{(l)} \right\rangle = 0, \forall\, \mathbf{v}^{(l)} \in P$ and hence,

$$\mathbf{x}^T A_{S\times S}\mathbf{x} = \mathbf{x}_S^T A_{S\times S}\mathbf{x}_S \geq -\tau \left\|\mathbf{x}_S\right\|^2 .$$

for $\tau \geq 0$. Now we examine the quadratic form for such vectors $\mathbf{x}$ in the subspace formed by vectors of set $T \setminus P$,

$$\begin{aligned}
\mathbf{x}^T Y \mathbf{x} &= \mathbf{x}^T \left(A_{S\times S} + A_{(V\setminus S)\times(V\setminus S)} + p\left(\mathbb{1}\mathbb{1}^T - \mathbb{1}_S\mathbb{1}_S^T - \mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right) + R + dI + B\right)\mathbf{x} \qquad (3.15)\\
&\geq \mathbf{x}_S^T \left(A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T\right)\mathbf{x}_S + \mathbf{x}_{V\setminus S}^T \left(A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right)\mathbf{x}_{V\setminus S} + (d - \|R\| - \|B\|)\left\|\mathbf{x}\right\|^2\\
&\geq \lambda_{\min}\underbrace{\left(A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T\right)}_{=T_1}\left\|\mathbf{x}_S\right\|^2 + \lambda_{\min}\underbrace{\left(A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right)}_{=T_2}\left\|\mathbf{x}_{V\setminus S}\right\|^2\\
&\qquad + \underbrace{(d - \|R\| - \|B\|)}_{=T_3}\left\|\mathbf{x}\right\|^2 .
\end{aligned}$$

Here, above while decomposing our matrix A we have used,

$$\begin{aligned}
A &= A_{S\times S} + A_{(V\setminus S)\times)V\setminus S)} + \mathbb{E}\left(A_{S\times(V\setminus S)} + A_{(V\setminus S)\times S}\right)\\
&\qquad + \left(A_{S\times(V\setminus S)+A_{(V\setminus S)\times S}} - \mathbb{E}\left(A_{S\times(V\setminus S)} + A_{(V\setminus S)\times S}\right)\right)\\
&= A_{S\times S} + p\left(\mathbb{1}\mathbb{1}^T - \mathbb{1}_S\mathbb{1}_S^T - \mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right) + R \quad \text{(Recall that } R \text{ is a matrix of form } M - \mathbb{E}[M]\text{)}
\end{aligned}$$

Consider the term T_1. Note that $-p\mathbb{1}_S\mathbb{1}_S^T$ shifts only the $\mathbb{1}_S$ eigenvector of $A_{S\times S}$ and the corresponding eigenvalue is then $d-pk$. Letting $D = A_{S\times S} - p\mathbb{1}_S\mathbb{1}_S^T$ and as long as $d-pk \geq -\tau$[12] we can claim that,

$$\mathbf{x}_S^T D \mathbf{x}_S \geq -\tau \left\|\mathbf{x}_S\right\|^2 . \qquad (3.16)$$

Now, take T_2. By our assumption on the smallest eigenvalue of $A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T$

[12]It is easy to check that for our choice of γ and τ this holds.

we have that,

$$\mathbf{x}_{V\setminus S}^T \left(A_{(V\setminus S)\times(V\setminus S)} - p\mathbb{1}_{V\setminus S}\mathbb{1}_{V\setminus S}^T\right)\mathbf{x}_{V\setminus S} \geq \lambda_{\min}\left(A_{(V\setminus S)\times(V\setminus S)}\right)\left\|\mathbf{x}_{V\setminus S}\right\|^2 \geq -\tau\left\|\mathbf{x}_{V\setminus S}\right\|^2 . \quad (3.17)$$

Using eq. (3.15), eq. (3.17) and eq. (3.16) we obtain that,

$$\mathbf{x}^T Y\mathbf{x} \geq (d - \tau - \|R\| - \|B\|)\|\mathbf{x}\|^2 .$$

Finally, for T_3, recall that $\|R\| \leq 2\sqrt{n}$ (Claim 2.1) and for our choice of τ and for $\|B\| \leq 256\sqrt{n(\log k)}/p^{5/2}$, in the regimes of k, n, p as stated, we have that,

$$d - \tau = d\left(1 - \frac{1/2-\alpha}{1/2+\alpha}\right) = \frac{2\alpha d}{1/2+\alpha} = 2\alpha pk \geq \frac{512}{p^{5/2}}\sqrt{n\log n} \text{ which is } \geq \|B\| + \|R\| .$$

Therefore, the dual matrix $Y \succeq 0$. Additionally, if the planted bipartite graph is connected, the vector $\mathbb{1}_{S_1} - \mathbb{1}_{S_2}$ is the only eigenvector with eigenvalue $-d$ and hence by our construction, it is the only eigenvector of Y with eigenvalue 0. Therefore we have that $\lambda_2(Y) > 0$. □

We next present the algorithm based on the guarantees provided by Lemma 3.6 as,

Algorithm 1:

Require: $G = (V, E)$, with a d-regular planted bipartite graph S ($d = \gamma pk, \gamma \geq 2/3$).

Ensure: The set of vertices in planted bipartite graph $\mathcal{S}$.

1: Initialize $\mathcal{S} = \phi$ to denote the set of recovered vertices.
2: Solve SDP 3.1.
3: Let $\mathcal{S} = \{i : \|\mathbf{x}_i\| > 0\}$
4: Return $\mathcal{S}$.

3.2.1.2 Pseudo-random values of dual variables

We now show that the dual SDP 3.2 is feasible. In particular this will be achieved by showing that eq. (3.14) has a solution. Also as discussed, we want the dual solution to satisfy the constraint $\|B\| = \mathcal{O}_p\left(\sqrt{n\log k}\right)$. Lemma 3.6 shows that if there exists a choice of dual variables B which satisfies this constraint, the dual SDP 3.2 is feasible and has rank $n-1$. Turns out, this constraint is implied if $B_{ij} \leq t = \mathcal{O}_p\left(\sqrt{\log k/k}\right)$ as shown below.

Lemma 3.7 *For $0 \leq B_{ij} \leq t$, $\forall (i,j) \in S \times (V \setminus S)$ we have that $\|B\|_2 \leq 2t\sqrt{k(n-k)}$.*

Proof:

$$\begin{aligned}\|B\|_2 &\leq \left\|B_{S,(V\setminus S)}\right\|_2 + \left\|B_{(V\setminus S),S}\right\|_2 = 2\left\|B_{S,(V\setminus S)}\right\|_2 \leq 2\left\|B_{S,V\setminus S}\right\|_F \qquad (\|M\|_2 \leq \|M\|_F)\\ &= 2\sqrt{\sum_{i\in S, j\in V\setminus S} B_{ij}^2} \leq 2t\sqrt{k\,(n-k)}\,.\end{aligned}$$

□

Corollary 3.1 *If* $B_{ij} \leq \left(32L_\tau/p^{3/2}\right)\sqrt{(\log k)/(1/2+\alpha)k}$ *we have* $\|B\|_2 \leq \left(256/p^{5/2}\right)\sqrt{n\log k}$.

Proof: Using Lemma 3.7 and setting $t = \left(32L_\tau/p^{3/2}\right)\sqrt{(\log k)/(1/2+\alpha)k}$ we have,

$$\|B\|_2 \leq 2t\sqrt{k\,(n-k)} = \frac{64L_\tau}{p^{3/2}\sqrt{1/2+\alpha}}\sqrt{n\log k}\,.$$

From Claim 3.1 we have that, $L_\tau \leq \gamma pk^2/2\tau^2$ and since we choose $\alpha \leq 1/6$, we have $\tau \geq d/2$ and since $\alpha > 0$ we have $\gamma \geq 1/2$. Using these bounds we obtain that $\|B\|_2 \leq \left(256/p^{5/2}\right)\sqrt{n\log k}$
□

Next we aim to show that there exists a solution to B_{ij}'s which satisfies eq. (3.14) and the criteria in Corollary 3.1 which eventually meets the hypothesis of Lemma 3.6.

Definition 3.3 *Consider the collection of linear systems in eq. (3.14). We define a collection of* spectral embedding based linear systems *by reorganizing this collection as follows.*

- *For* $j \in V\setminus S$*, define a system of equations* $\mathcal{F}_j$*.*
- *In all, this gives a collection of systems* $\{\mathcal{F}_j\}_{j\in V\setminus S}$*. Each system contains* $L_\tau \times |S|$ *variables. In particular, the system* $\mathcal{F}_j$ *is expressed in the standard form* $W\mathbf{x} = \mathbf{b}$*, where* $W \in \mathbb{R}^{L_\tau\times k}$ *is a matrix formed by stacking the vectors* $\mathbf{v}^{(l)} \in P$ *as rows.*

Formally a vector $\mathbf{w}^{(i)} \in \mathbb{R}^{L_\tau}$ is defined such that, $w_l^{(i)} = v_i^{(l)}$. This viewpoint goes by the name of *spectral embedding* in literature and has been explored in other works on graph partitioning, [NJW01, LOT12, LRTV12] etc.

Fix $j \in V\setminus S$ and consider the system $\mathcal{F}_j$. The vector $\mathbf{b}$ in this system is a row vector of size $L_\tau \times 1$ and has entries given by $b_l = -\sum_{i\in S} A_{ij}v_i^{(l)}, \forall l \in [L_\tau]$ and $\mathbf{x}$ here is a row vector of size $k\times 1$ where the entry $x_i = B_{ij}$ (recall that we have fixed a $j \in V\setminus S$). However since B_{ij}'s are not arbitrary variables but dual variables of SDP 3.2, they are constrained. Firstly, they should only be defined for $i \in N(j)$ and secondly they are required to be non-negative. Since the graph on $S\times(V\setminus S)$ is random, the choice of random edges while choosing $N(j)$ (in model

construction, refer Definition 3.1) fixes those corresponding B_{ij}'s that have to be set to zero (whenever the edge $\{i, j\}$ is not present in the graph). For simplicity, consider the case $p = 1$, i.e. when $A_{ij} = 1$ for all $i \in S, j \notin S$. Here the full set of B_{ij}'s in $S \times (V \setminus S)$ are available and it is easy to satisfy eq. (3.14). This is because for any vector $\mathbf{y} \in \mathbb{R}^L$,

$$\mathbf{b}^T\mathbf{y} = \sum_{r\in[L]} b_r y_r = -\sum_{r\in[L]}\sum_{i\in S} v_i^{(r)} y_r = -\sum_{i\in S}\sum_{r\in[L]} v_i^{(r)} y_r \tag{3.18}$$

$$= -\sum_{i\in S}\sum_{r\in[L]} w_r^{(i)} y_r = -\sum_{i\in S}\langle \mathbf{w}^{(i)}, y\rangle = -\left\langle \sum_{i\in S}\mathbf{w}^{(i)}, \mathbf{y}\right\rangle = 0\,. \tag{3.19}$$

where the last equality holds since $\mathbf{v}^{(l)}$ is orthogonal to $\mathbb{1}_S$ eigenvector[13] translates to $\sum_{i\in S}\mathbf{w}^{(i)}$ beging zero. Using the standard variant of Farkas' Lemma (refer [BV04]), this immediately implies the existence of a solution to equation eq. (3.14). However, in general, for $p < 1$, the eq. (3.18) does not hold and we need to do more work here.

Now, let $\tilde{W}$ denote the submatrix after removing the columns corresponding to $i \notin N(j)$ and t to be the absolute bound on the entries of B matrix (as desired in Corollary 3.1). We thus consider the following feasibility LP formulation for this problem.

LP 3.1

$$\tilde{W}\mathbf{x} = \mathbf{b} \tag{3.20}$$

$$0 \le \mathbf{x} \le t\mathbb{1}\,. \tag{3.21}$$

The feasibility for such LPs is typically characterized by the Theorem of Alternatives (e.g., Farkas' Lemma). The standard variants for these deal with either the equality constraints or the inequality constraints. Here, our LP 3.1 has mixed constraints, but we can derive a Farkas' Lemma style Theorem of Alternatives (along the lines of [BV04]) as.

Proposition 3.1 (Folklore) *For a fixed $\mathbf{u} \in \mathbb{R}^{L_\tau}$, exactly one out of these two systems of linear equations is feasible,*

1. $\{\mathbf{x} : C\mathbf{x} = \mathbf{f},\ \mathbf{0} \le \mathbf{x} \le \mathbf{u}\}$.
2. $\left\{\mathbf{y} : C^T\mathbf{y} + \mathbf{z} \ge 0, \mathbf{f}^T\mathbf{y} + \mathbf{u}^T\mathbf{z} < 0, \mathbf{y} \in \mathbb{R}^{L_\tau}, \mathbf{z} \ge 0\right\}$.

[13] $\mathbb{1}_S$ vector has eigenvalue d and hence is not a threshold rank eigenvector.

Proof: For completeness, we give a proof along the lines of proof for the standard variants of Farkas' Lemma in Section 3.3.2. □

Corollary 3.2 *The primal LP 3.1 is feasible iff*

$$\forall \mathbf{y} \in \mathbb{R}^{L_\tau}, \forall \mathbf{z} \geq 0, \tilde{W}^T\mathbf{y} + \mathbf{z} \geq 0 \implies \mathbf{b}^T\mathbf{y} + t\,\langle \mathbf{z}, \mathbb{1} \rangle \geq 0\,. \tag{3.22}$$

Proof: The above follows by setting $\mathbf{u} = t\mathbb{1}$, $\mathbf{f} = \mathbf{b}$ and $C = \tilde{W}$ in Proposition 3.1. □

We wish to compute a value of $t > 0$ such that the eq. (3.22) holds. Proving this seems to require a better understanding of the structure of these embedding vectors $\mathbf{w}^{(i)}$'s. Since these are intimately connected to the threshold rank eigenvectors $\mathbf{v}^{(l)}$, we use the structure of *threshold rank eigenvectors* to characterize the embedding vectors. Therefore, next (in Lemma 3.8 and Lemma 3.9) we prove some useful properties of these eigenvectors which will be used in the analysis. Throughout the rest of the section we will assume that the eigenvectors $\left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(L_\tau)}\right\} \in P$ have unit norm.

Lemma 3.8 *For an eigenvector $\mathbf{v}^{(l)} \in P$ we have that, $\left\|\mathbf{v}^{(l)}\right\|_\infty \leq \dfrac{2}{\sqrt{d}}$.*

Proof: Since $\mathbf{v}^{(l)}$ is an eigenvector of $A|_{S\times S}$ we have that,

$$A|_{S\times S}\,\mathbf{v}^{(l)} = \lambda_l \mathbf{v}^{(l)} = -d(1-\delta)\mathbf{v}^{(l)}\,. \tag{3.23}$$

where $\delta \in [0, 1/2]$ (since by our choice of α we always have $\tau > d/2$). We compare the vectors in eq. (3.23) component wise and we have that,

$$\sum_{i \in N(j)} \mathbf{v}_i^{(l)} = -d(1-\delta)\mathbf{v}_j^{(l)} \quad \forall j \in S\,. \tag{3.24}$$

We take absolute value on both sides and use Cauchy–Schwarz inequality. This gives

$$\left|-d(1-\delta)\mathbf{v}_j^{(l)}\right| = \left|\sum_{i\in N(j)} \mathbf{v}_i^{(l)}\right| = \left|\left\langle \mathbb{1}_{N(j)}, \mathbf{v}^{(l)} \right\rangle\right| \leq \sqrt{d}\sqrt{\left\|\mathbf{v}^{(l)}\right\|^2} \leq \sqrt{d}\,.$$

This allows us to give an upper bound on the l_∞ norm of the eigenvector $\mathbf{v}^{(l)}$ by comparing the left and right hand side as,

$$\left|v_j^{(l)}\right| \leq \frac{1}{(1-\delta)\sqrt{d}} \leq \frac{2}{\sqrt{d}}\,. \tag{3.25}$$

□

Lemma 3.9 *For a fixed $j \in V \setminus S$ and for $\left\{\mathbf{w}^{(i)} | i \in N(j)\right\}$, the spectral embedding of randomly selected neighbours of j, we have that with probability $\geq 1 - \mathcal{O}\left(1/k^4\right)$*

$$\left\| \sum_{i \in N(j)} \mathbf{w}^{(i)} \right\|_2 \leq 2\sqrt{L_\tau \log k}\,.$$

Proof: To show the claim above, we first show that $\left\|\sum_{i\in N(j)} \mathbf{w}^{(i)}\right\|_\infty \leq 2\sqrt{\log k}$. Now since $\left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(L_\tau)}\right\}$ were orthogonal to $\mathbb{1}_S$ vector to start with, $\sum_{i\in S} \mathbf{w}^{(i)} = 0$. Let $r \in [L_\tau]$ denote the index which maximizes $\left|\sum_{i\in N(j)} w_l^{(i)}\right|$. Since $\sum_{i\in S} w_r^{(i)} = 0$ we have that,

$$\mathbb{E}\left[\sum_{i\in N(j)} w_r^{(i)}\right] = 0\,.$$

Then we can use Hoeffding bounds (Fact 2.2) to bound as,

$$\begin{aligned}\mathbb{P}\left[\left|\sum_{i\in N(j)} w_r^{(i)}\right| \geq 2\sqrt{\log k}\right] &\leq 2\exp\left(-\frac{4\log k}{\sum_{i=1}^k {w_r^{(i)}}^2}\right) = 2\exp\left(-\frac{4\log k}{\sum_{i=1}^k {v_i^{(r)}}^2}\right) \\ &= 2\exp\left(-4\log k\right) = \frac{2}{k^4}\,.\end{aligned}$$

Therefore with probability $\geq 1 - \mathcal{O}\left(1/k^4\right)$, we have that $\left\|\sum_{i\in N(j)} \mathbf{w}^{(i)}\right\|_2 \leq 2\sqrt{L_\tau \log k}\,.$ □

Proposition 3.2 *For choice of $t = \left(32L_\tau/p^{3/2}\right)\left(\sqrt{\left(\log k\right)/\gamma k}\right)$ and for $p \geq 5\left(\left(\log k\right)/\gamma^4 k\right)^{1/6}$, with high probability (over the randomness of the input instance),*

$$\forall\, \mathbf{y} \in \mathbb{R}^{L_\tau}, \forall\, \mathbf{z} \geq 0, \tilde{W}^T\mathbf{y} + \mathbf{z} \geq 0 \implies \mathbf{b}^T\mathbf{y} + t\left\langle \mathbf{z}, \mathbb{1}\right\rangle \geq 0\,.$$

Proof: For the sake of contradiction, suppose there exists a $\mathbf{y} \in \mathbb{R}^{L_\tau}$ and $\mathbf{z} \geq 0$ such that $\tilde{W}^T\mathbf{y} + \mathbf{z} \geq 0$ and $\mathbf{b}^T\mathbf{y} + t\left\langle \mathbf{z}, \mathbb{1}\right\rangle < 0$. We can then write the given condition

$\tilde{W}^T\mathbf{y} + \mathbf{z} \geq 0$ in the form $\left(\tilde{W}^T\mathbf{y}\right)_i + z_i \geq 0$, use $\left(\tilde{W}^T\mathbf{y}\right)_i = \left\langle \mathbf{w}^{(i)}, \mathbf{y}\right\rangle$ to get $z_i \geq -\left\langle \mathbf{w}^{(i)}, \mathbf{y}\right\rangle$.

Now since we have that $\mathbf{z} \geq 0$, it is sufficient if we prove our claim for the minimum possible value of $\mathbf{z}$. We do so by setting $z_i = \max\left\{0, -\left\langle \mathbf{w}^{(i)}, \mathbf{y}\right\rangle\right\}$. This is sufficient because $t > 0$ and

$\langle \mathbf{z}, \mathbb{1} \rangle \geq 0$ and hence $\mathbf{b}^T\mathbf{y} + t\langle \mathbf{z}, \mathbb{1} \rangle$ can only increase for a larger value of $\mathbf{z}$. We then express the term $\mathbf{b}^T\mathbf{y}$ as,

$$\mathbf{b}^T\mathbf{y} = \sum_{r\in[L_\tau]} b_r y_r = -\sum_{r\in[L_\tau]}\sum_{i\in N(j)} v_i^{(r)} y_r = -\sum_{i\in N(j)}\sum_{r\in[L_\tau]} v_i^{(r)} y_r \tag{3.26}$$

$$= -\sum_{i\in N(j)}\sum_{r\in[L_\tau]} w_r^{(i)} y_r = -\sum_{i\in N(j)} \left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle . \tag{3.27}$$

Using the value of $\mathbf{z}$ as above i.e $z_i = \max\left\{0, -\left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle\right\}$, we can rewrite the condition $\mathbf{b}^T\mathbf{y} + t\langle \mathbf{z}, \mathbb{1} \rangle < 0$ as,

$$\sum_{i\in N(j)} \left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle + t \sum_{i\in N(j)} \min\left\{0, \left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle\right\} > 0 . \tag{3.28}$$

To finish the proof by contradiction, we will show that eq. (3.28) does not hold for our choice of t. Without loss of generality we assume $\|\mathbf{y}\| = 1$ and proceed to bound the first term in the expression on left hand side of eq. (3.28) as,

$$\sum_{i\in N(j)} \left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle = \left\langle \mathbf{y}, \sum_{i\in N(j)} \mathbf{w}^{(i)} \right\rangle \leq \|\mathbf{y}\| \left\| \sum_{i\in N(j)} \mathbf{w}^{(i)} \right\| \leq 2\sqrt{L_\tau \log k} \tag{3.29}$$

where the last inequality follows from Lemma 3.9. In Lemma 3.11 we give an upper bound on the second term as

$$\sum_{i\in N(j)} \min\left\{0, \left\langle \mathbf{w}^{(i)}, \mathbf{y} \right\rangle\right\} \leq -(p/16)\sqrt{d/L_\tau} . \tag{3.30}$$

The bounds in eq. (3.29) and eq. (3.30) hold for a fixed $j \in V\setminus S$, but since these bounds come from Lemma 3.9 and Lemma 3.11 which hold with probability $\geq 1 - \mathcal{O}\left(1/n^2\right)$; we do a union bound over all $j \in V\setminus S$ such that these bounds hold for all $j \in V\setminus S$ with high probability (over the randomness of the input instance). Then for our choice of $t = \left(32L_\tau/p^{3/2}\right)\left(\sqrt{(\log k)/\gamma k}\right)$ in eq. (3.28) we get,

$$2\sqrt{L_\tau \log k} - \left(\frac{32L_\tau\sqrt{\log k}}{p^{3/2}\sqrt{\gamma k}}\right)\left(\frac{p\sqrt{d}}{16\sqrt{L_\tau}}\right) = 2\sqrt{L_\tau \log k} - \left(\frac{32L_\tau\sqrt{\log k}}{p^{3/2}\sqrt{\gamma k}}\right)\left(\frac{p^{3/2}\sqrt{\gamma k}}{16\sqrt{L_\tau}}\right) = 0 .$$

and hence eq. (3.28) doesn't hold.

□

Towards bounding the second term, we use a well known fact (e.g. shown in [LRTV11])) that these *spectral embedding vectors* are isotropic (upto a scaling of $\sqrt{k}$) where $i \in N(j)$ are being sampled randomly as per $G_{n,p}$ distribution. For $p = 1$, we can then show that equation eq. (3.28) does not hold and we are done. However, for $p < 1$, we have $\mathbb{E}\left[\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right] = p$ and these vectors are close to isotropic. We formalize this in lemma 3.10 by using Matrix Bernstein concentration inequality. We thus show a lower bound on

$$\sum_{i \in N(j)} \left\langle \mathbf{y}, \mathbf{w}^{(i)} \right\rangle^2 \geq p/2\,. \tag{3.31}$$

showing that these are $p/2$-isotropic[14].

Lemma 3.10 *For an arbitrary $\mathbf{y} \in \mathbb{R}^{L_\tau}$ and a fixed $j \in V \setminus S$, for $p \geq 5\left(\log k/\gamma^4 k\right)^{1/6}$, we have that with probability $\geq 1 - \mathcal{O}\left(1/k^4\right)$,*

$$\sum_{i \in N(j)} \left\langle \mathbf{y}, \mathbf{w}^{(i)} \right\rangle^2 \geq \frac{p}{2}\,.$$

Proof:

$$\sum_{i \in N(j)} \left\langle \mathbf{y}, \mathbf{w}^{(i)} \right\rangle^2 = \sum_{i \in N(j)} \left(\mathbf{y}^T \mathbf{w}^{(i)}\right)\left(\mathbf{w}^{(i)^T}\mathbf{y}\right) = \mathbf{y}^T \left(\sum_{i \in N(j)} \mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right)\mathbf{y}\,.$$

Claim 3.2 *We let $M = \sum_{i \in N(j)} \mathbf{w}^{(i)}\mathbf{w}^{(i)^T}$, a matrix of size $L_\tau \times L_\tau$ then,*

$$\mathbb{E}\left[M_{rs}\right] = 0 \text{ for } r \neq s \text{ and } \mathbb{E}\left[M_{rr}\right] = p\,.$$

Proof: A proof of this can be found in [LRTV11]. For completeness we give a proof in Section 3.3.3 □

Therefore we conclude that $\mathbb{E}[M] = pI$. Next, we show the concentration of the matrix M by using Matrix Bernstein inequality Theorem 6.1 [Tro12] stated here as,

Fact 3.2 (Matrix Bernstein inequality) *For a sequence of independent,symmetric, and random matrices $\left\{X_i \in \mathbb{R}^{k \times k}\right\}_{i=1}^{k}$ where,*

- $\mathbb{E}\left[X_i\right] = 0$

[14] We say that a distribution is C-isotropic if the eigenvalues of covariance matrix lie in $[(1/C), C]$.

- $\|X_i\| \leq \rho$
- $\nu = \|\mathbb{E}\left[\sum_i (X_i^2)\right]\|$

we have that for all $\varepsilon \geq 0$,

$$\mathbb{P}\left[\left\|\sum_i X_i\right\| \geq \varepsilon\right] \leq 2k \exp\left(\frac{-\varepsilon^2}{2\nu + 2\rho 3}\right) .$$

In our setting we let,

$$Y_i = \begin{cases} \mathbf{w}^{(i)}\mathbf{w}^{(i)^T} & \text{if } i \in N(j) \\ 0 & \text{otherwise .} \end{cases}$$

and define the random matrices $X_i = Y_i - \mathbb{E}[Y_i]$ so that $\mathbb{E}[X_i] = 0$. Writing this explicitly we have,

$$X_i = \begin{cases} (1-p)\mathbf{w}^{(i)}\mathbf{w}^{(i)^T} & \text{if } i \in N(j) \\ -p\mathbf{w}^{(i)}\mathbf{w}^{(i)^T} & \text{otherwise .} \end{cases}$$

From above we can see that,

$$\|X_i\| \leq \max\left\{\left\|p\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right\|, \left\|(1-p)\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right\|\right\} \leq \left\|\mathbf{w}^{(i)}\right\|^2 \leq \frac{2L_\tau}{d}$$

where we have used the fact that $\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}$ is a rank one matrix and hence $\left\|\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right\| \leq \left\|\mathbf{w}^{(i)}\right\|^2$. and the last inequality holds due to lemma 3.8. Hence we choose value of $\rho = 2L_\tau/d$. Next we bound the variance ν as,

$$\nu = \left\|\mathbb{E}\left[\sum_{i\in S} X_i^2\right]\right\| = \left\|\sum_{i\in S}\mathbb{E}\left[X_i^2\right]\right\| \leq \sum_{i\in S}\left\|\mathbb{E}\left[X_i^2\right]\right\| . \tag{3.32}$$

To compute $\mathbb{E}[X_i^2]$ we note that,

$$X_i^2 = \begin{cases} (1-p)^2\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\mathbf{w}^{(i)}\mathbf{w}^{(i)^T} & \text{if } i \in N(j) \\ p^2\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\mathbf{w}^{(i)}\mathbf{w}^{(i)^T} & \text{otherwise .} \end{cases}$$

Therefore $\mathbb{E}\left[X_i^2\right] = p(1-p)\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}$ and using this in eq. (3.32) we get,

$$\nu \leq \sum_{i\in S}\left\|\mathbb{E}\left[X_i^2\right]\right\| \leq p(1-p)\sum_{i\in S}\left\|\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right\|\left\|\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right\| \leq \sum_{i\in S}\left\|\mathbf{w}^{(i)}\right\|^4 = k\left\|\mathbf{w}^{(i)}\right\|^4 \leq \frac{16{L_\tau}^2 k}{d^2}\,. \tag{3.33}$$

Using the value of ν, ρ and the fact that $\mathbb{E}\left[\sum_{i\in N(j)}\mathbf{w}^{(i)}\mathbf{w}^{(i)^T}\right] = pI$ and choosing $\varepsilon = p/2$ we have that,

$$\begin{aligned}
\mathbb{P}\left[\left\|\sum_{i\in N(j)}\mathbf{w}^{(i)}\mathbf{w}^{(i)^T} - pI\right\| \geq \frac{p}{2}\right] &\leq 2k\exp\left(-\frac{(p^2/4)}{2\left(16{L_\tau}^2 k/d^2\right) + (2pL_\tau/3d)}\right) \qquad (3.34)\\
&\leq 2k\exp\left(-\frac{p^2/4}{2\left(32L_\tau^2 k\right)/d^2}\right) \qquad \left(\left(32L_\tau^2 k\right)/d^2 \geq \frac{(2pL_\tau)}{3d}\right)\\
&\leq 2k\exp\left(-\frac{\gamma^4 p^6 k}{1024}\right) \qquad \text{(Using Claim 3.1)}\,.
\end{aligned}$$

For $\gamma \geq 1/2$ and $p \geq 5\left(\log k/\gamma^4 k\right)^{1/6}$, Lemma 3.10 holds with probability $\geq 1 - \mathcal{O}\left(1/k^4\right)$. □

With these bounds at hand we proceed to bound the second term in eq. (3.28) as,

Lemma 3.11 *For a unit vector $\mathbf{y} \in \mathbb{R}^{L_\tau}$ and $j \in V \setminus S$ and for $p \geq 5\left(\left(\log k\right)/\gamma^4 k\right)^{1/6}$ with probability $\geq 1 - \mathcal{O}\left(1/k^4\right)$ we have that,*

$$\sum_{i\in N(j)}\min\left\{0, \left\langle\mathbf{w}^{(i)}, \mathbf{y}\right\rangle\right\} \leq -\frac{p}{16}\sqrt{\frac{d}{L_\tau}}\,.$$

Proof: We start by defining these two sets,

$$\mathcal{P} = \left\{i \in N(j) : \left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle \geq 0\right\}$$

$$\mathcal{N} = \left\{i \in N(j) : \left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle < 0\right\}$$

Clearly the summation over terms in $\mathcal{P}$ is 0 so we focus on the terms in $\mathcal{N}$. We first consider the case when $\sum_{i\in\mathcal{N}}\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle^2 \geq p/4$ and since $\left|\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle\right| \leq \|\mathbf{y}\|\left\|\mathbf{w}^{(i)}\right\| \leq 2\sqrt{L\tau}/\sqrt{d}$ (using Lemma 3.8) we have that,

$$\frac{p}{4} \leq \sum_{i\in\mathcal{N}}\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle^2 = \sum_{i\in\mathcal{N}}\left|\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle\right|^2 \leq 2\sqrt{\frac{L_\tau}{d}}\sum_{i\in\mathcal{N}}\left|\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle\right|\,.$$

and therefore $\sum_{i\in\mathcal{N}}\left\langle\mathbf{y}, \mathbf{w}^{(i)}\right\rangle \leq -\left(p/8\right)\sqrt{d/L_\tau}$.

Next we consider the case when $\sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle^2 < p/4$. Now in this case, from Lemma 3.9 we know that, $\left\| \sum_{i\in N(j)} \mathbf{w}^{(i)} \right\|_2 \leq 2\sqrt{L_\tau \log k}$, we can write,

$$\left| \sum_{i\in N(j)} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle \right| = \left| \left\langle \mathbf{y}, \sum_{i\in N(j)} \mathbf{w}^{(i)} \right\rangle \right| \leq \|\mathbf{y}\| \left\| \sum_{i\in N(j)} \mathbf{w}^{(i)} \right\| \leq 2\sqrt{L_\tau \log k}\,. \tag{3.35}$$

From eq. (3.35) we obtain that,

$$\sum_{i\in\mathcal{P}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle \leq -\sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle + 2\sqrt{L_\tau \log k}\,.$$

Using eq. (3.35) and $\sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle^2 < p/4$ we have that,

$$\frac{p}{4} \leq \sum_{i\in\mathcal{P}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle^2 \leq \|\mathbf{y}\| \max_{i\in\mathcal{P}} \|\mathbf{w}^{(i)}\| \sum_{i\in\mathcal{P}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle \leq 2\sqrt{\frac{L_\tau}{d}} \sum_{i\in\mathcal{P}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle \tag{3.36}$$

$$\leq -2\sqrt{\frac{L_\tau}{d}} \sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle + 4\sqrt{\frac{{L_\tau}^2 \log k}{d}}\,. \tag{3.37}$$

We wish that the term $4\sqrt{{L_\tau}^2 (\log k)/d} \leq p/8$ and by using L_τ from Claim 3.1, we get that this holds is implied by

$$p^2 \geq 1024 \left(\frac{\gamma p k^2}{2\tau^2}\right)^2 \frac{\log k}{d} \text{ which holds if } p \geq 1024 \left(\frac{\gamma p k^2}{2(d/2)^2}\right)^2 \frac{\log k}{d}$$

where we have accounted for the worst case value of $\tau = d/2$. Now using $d = \gamma p k$ we can rewrite as,

$$p^2 \geq 1024 \left(\frac{\gamma p k^2}{2(\gamma p k/2)^2}\right)^2 \frac{\log k}{\gamma p k} = \left(\frac{4096}{\gamma^3 p^3}\right) \frac{\log k}{k}\,.$$

Therefore when $p \geq 6\,(\log k/\gamma^3 k)^{1/5}$ we have $4\sqrt{{L_\tau}^2 (\log k)/d} \leq p/8$ and solving for $\sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle$ in eq. (3.36) we have that,

$$\sum_{i\in\mathcal{N}} \langle \mathbf{y}, \mathbf{w}^{(i)} \rangle \leq -\frac{p}{16}\sqrt{\frac{d}{L_\tau}}\,.$$

We note that this lower bound on p is subsumed by the bound from Lemma 3.10. □

3.2.2 Low degree regimes

Now we consider the regimes when $d = \gamma pk$ with $\gamma \leq 2/3$. We next show that a simple algorithm (due to [Kuc95]) that collects the bottom k degrees of the graph will work in these regimes since the vertices in S will have degrees standing out from the rest of the graph.

Algorithm 2:

Require: $G = (V, E)$, with a d-regular planted bipartite graph S ($d = \gamma pk, \gamma \leq 2/3$).

Ensure: The set of vertices in planted bipartite graph $\mathcal{S}$.

1: Initialize $\mathcal{S} = \phi$ to denote the set of recovered vertices.
2: Sort the degrees of the vertices in G.
3: Let $\mathcal{S}$ be the set of bottom k degrees after sorting.
4: Return $\mathcal{S}$.

Lemma 3.12 *For $k \geq 6\sqrt{n \log n}/p$ the Algorithm 2 returns the planted set S with high probability (over the randomness of the input).*

Proof: For a vertex $v \in S$ the degree of the vertex can be upper bounded with high probability (over the randomness of the input) as,

$$d(v) \leq d + p(n-k) + \sqrt{n \log n}\,.$$

and for a vertex $v' \notin S$ the degree can be lower bounded with high probability (over the randomness of the input) as,

$$d(v') \geq pn - \sqrt{n \log n}\,.$$

where the $\sqrt{n \log n}$ terms are the high probability bounds after applying Chernoff bounds on respective degrees. The degrees differ as,

$$d(v') - d(v) \geq pk - d - 2\sqrt{n \log n} = \frac{pk}{3} - 2\sqrt{n \log n}\,. \tag{3.38}$$

where the equality follows from choice of $\tau = 2d/3$. It is also evident from eq. (3.38) that for $k \geq 6\sqrt{n \log n}/p$, with high probability (over the randomness of the input), the degree for vertex $v \in S$ stands out. □

3.2.3 Action of Adversary

We recall that in Item 5 of our model construction (Definition 3.1), we allow a monotone adversary to arbitrarily add edges between two vertices of $(V \setminus S) \times (V \setminus S)$. Next, we argue

that despite the action of an adversary, our algorithms. (Algorithm 1 and Algorithm 2) still return the planted set S. For the regimes of $\gamma \leq 2/3$ where Algorithm 2 returns the planted set, this is obvious since our algorithm relies on the difference between the degree of vertices in S and $V \setminus S$ and the action of adversary only amplifies the difference. For $\gamma \geq 2/3$, we use the argument from the work [FK01] to recover the planted set S under the action of an adversary.

Lemma 3.13 *Let $\tilde{G}$ be the graph obtained after action of adversary on G as per Item 5, $\mathsf{OPT}\left(\tilde{G}\right) = \mathsf{OPT}(G)$ and the solution in eq. (3.6) is the unique solution.*

Proof: We start by writing out an exact formulation of the problem as an integer quadratic program,

QP 3.1

$$\min \sum_{\{i,j\} \in E} x_i x_j$$

subject to

$$\sum_{i \in V} x_i^2 = k \tag{3.39}$$

$$x_i x_j \leq 0 \qquad \forall \{i,j\} \in E \tag{3.40}$$

$$x_i^2 \in \{0,1\} \ . \tag{3.41}$$

We denote by $\mathsf{QPOPT}(G)$ to be the minimum objective value to QP 3.1. Now since this is an exact formulation for our problem we have that, $\mathsf{OPT}(G) = \mathsf{QPOPT}(G)$. Note that SDP 3.1 is a relaxation to QP 3.1 (after scaling by a factor of k) and hence,

$$\mathsf{SDPOPT}(G) \leq \mathsf{OPT}(G) = \mathsf{QPOPT}(G) \ . \tag{3.42}$$

We will prove our claim by induction on the number of edges added by adversary in Item 5. We consider the base case first and let G' to be the graph obtained after adding an edge to G, then since the integral solution Equation (3.6) is still a feasible solution to QP 3.1 for G', using equality in eq. (3.42) we have that,

$$\mathsf{OPT}(G') = \mathsf{QPOPT}(G') \leq -kd \, .$$

Also using inequality in eq. (3.42) we have that,

$$-kd - 1 \leq \mathsf{SDPOPT}(G') \leq \mathsf{OPT}(G') \, ,$$

where first inequality follows from the fact that a solution of value strictly smaller than $-kd-1$ to $\mathsf{SDPOPT}(G')$ would imply a solution of value strictly smaller that $-kd$ to $\mathsf{SDPOPT}(G)$ since removing an edge, the objective falls by at most 1, due to the SDP constraints. Therefore we have that,

$$-kd-1 \leq \mathsf{OPT}(G') \leq -kd\,.$$

Now since $\mathsf{OPT}(G')$ is an integer, it can either be $-kd$ or $-kd-1$. However if $\mathsf{OPT}(G') = -kd-1$, then $\mathsf{SDPOPT}(G') \leq -kd-1$ and once we remove the edge back from G' to obtain G the SDP solution is still a feasible solution to G with value less than or equal to $-kd$. However this solution is different from the integral solution $\mathbf{g}\mathbf{g}^T$ since for the solution $\mathbf{g}\mathbf{g}^T$, $\mathsf{SDPOPT}(G')$ would have been $-kd$ as well. Therefore, we have obtained a new solution to our *SDP* 3.1. However as argued in Fact 2.6, our SDP can only have a unique solution. Therefore we have that $\mathsf{OPT}(G') = -kd = \mathsf{OPT}(G)$. This argument also proves that the SDP solution to G' has to be $\mathbf{g}\mathbf{g}^T$ since otherwise we again get a contradiction to Fact 2.6 for G. Now $\tilde{G}$ is obtained by a sequence of such operations on G and the argument above holds for each such operation, and therefore,

$$\mathsf{OPT}(G) = \mathsf{OPT}(G') = \ldots = \mathsf{OPT}\left(\tilde{G}\right).$$

□

We have now established everything to prove Theorem 3.4.

Proof: [Proof of Theorem 3.4] For $\gamma \leq 2/3$, as argued above in Lemma 3.12 that the Algorithm 2 recovers the planted set. For $\gamma \geq 2/3$, we know that for $p \geq 5\left((\log k)/\gamma^4 k\right)^{1/6}$ and choice of $t = \left(32L_\tau/p^{3/2}\right)\left(\sqrt{\log k/\gamma k}\right)$ the LP 3.1 has a solution. As argued in Corollary 3.1 this already implies that $\|B\|_2 \leq \left(256/p^{5/2}\right)\sqrt{n\log k}$.

Given this bound on $\|B\|_2$, Lemma 3.6 guarantees that the Algorithm 1 returns the planted set S with high probability (over the randomness of the input). Furthermore the lower bound on k in Lemma 3.6 implies the bound on k in the Lemma 3.12. As argued in Lemma 3.13, the solution due to SDP remains optimal even after action of monotone adversary as per Item 5. Therefore Algorithm 1 and Algorithm 2 still return the planted set S respectively. □

3.3 Miscellaneous proofs

3.3.1 Computing the dual of SDP 3.1

We compute the dual by writing SDP in matrix form as,

SDP 3.3

$$\min \langle A, X \rangle$$

subject to

$$\langle I, X \rangle = 1 \tag{3.43}$$
$$\langle \mathbb{1}_{ij}, X \rangle \leq 0 \qquad \forall \{i,j\} \in E \tag{3.44}$$
$$\langle \mathbb{1}_{ji}, X \rangle \leq 0 \qquad \forall \{i,j\} \in E \tag{3.45}$$
$$X \succeq 0\,. \tag{3.46}$$

and we compute the Lagrangian associated with SDP 3.3 as,

$$\mathcal{L}(X, B, Y, \beta) = \langle A, X \rangle + \beta(1 - \langle I, X \rangle) + \sum_{\{i,j\} \in E} B_{ij} \langle \mathbb{1}_{ij}, X \rangle + \sum_{\{i,j\} \in E} B_{ji} \langle \mathbb{1}_{ji}, X \rangle - \langle Y, X \rangle$$

where β is unconstrained Lagrange dual variable for the equality constraint and B is the matrix of non-negative Lagrange variables B_{ij}'s and B_{ji}'s for the inequality constraints and Y is a p.s.d Lagrange dual matrix. We then compute the corresponding Lagrange dual function as,

$$g(Y, B, \beta) = \inf_{X \succeq 0} \mathcal{L}(X, B, Y, \beta) = \begin{cases} \beta & \text{if } A - \beta I + B - Y = 0 \\ -\infty & \text{otherwise} \end{cases}$$

Therefore we obtain the dual SDP program as described in SDP 3.2

3.3.2 Proof of Proposition 3.1

Proof: Let there be a solution to the first system of equations $\{\mathbf{x} : C\mathbf{x} = \mathbf{f},\ 0 \leq \mathbf{x} \leq \mathbf{u}\}$ then,

$$\mathbf{f}^T\mathbf{y} + \mathbf{u}^T\mathbf{z} = (C\mathbf{x})^T\mathbf{y} + \mathbf{u}^T\mathbf{z} = \mathbf{x}^T(C^T\mathbf{y}) + \mathbf{u}^T\mathbf{z} = \mathbf{x}^T(C^T\mathbf{y} + \mathbf{z}) + \mathbf{z}^T(\mathbf{u} - \mathbf{x}) \geq 0\,.$$

and hence the second system of equations does not have a solution.
To show that the system $\left\{\mathbf{y} : C^T\mathbf{y} + \mathbf{z} \geq 0, \mathbf{f}^T\mathbf{y} + \mathbf{u}^T\mathbf{z} < 0, \mathbf{y} \in \mathbb{R}^L, \mathbf{z} \geq 0\right\}$ has a solution implies that the first system has a solution; we start with a contradiction. Let there be no solution to $\{\mathbf{x} : C\mathbf{x} = \mathbf{f},\ 0 \leq \mathbf{x} \leq \mathbf{u}\}$. In other words

$$\nexists \mathbf{f} \in \mathcal{C} \text{ where } \mathcal{C} := \{C\mathbf{x} : 0 \leq \mathbf{x} \leq \mathbf{u}\}\ .$$

Since $\mathcal{C}$ is a closed convex set; by hyperplane separation theorem we have that there exists a $\mathbf{y}$

such that,

$$\langle \mathbf{y}, \mathbf{f} \rangle > \sup_{\mathbf{c} \in \mathcal{C}} \langle \mathbf{y}, \mathbf{c} \rangle = \sup_{0 \leq \mathbf{x} \leq \mathbf{u}} \langle \mathbf{y}, C\mathbf{x} \rangle = \sup_{0 \leq \mathbf{x} \leq \mathbf{u}} \left\langle C^T \mathbf{y}, \mathbf{x} \right\rangle > 0 \,.$$

where the last inequality holds by setting $\mathbf{x} = 0$ since $\mathbf{0} \in \mathcal{C}$. Now since both $\mathbf{u}, \mathbf{z} \geq \mathbf{0}$ we have that,

$$\mathbf{f}^T \mathbf{y} + \mathbf{u}^T \mathbf{z} > 0 \,.$$

However this is a contradiction to the fact that this system had a solution. Hence the first system has no solution. □

3.3.3 Proof of Claim 3.2

Proof: By our definition of M we have that,

$$M_{rr} = \sum_{i \in N(j)} \left(\mathbf{w}^{(i)} \mathbf{w}^{(i)^T} \right)_{rr} = \sum_{i \in N(j)} \left(w_r^{(i)} \right)^2$$

and taking expectation over the choice of random edges $i \in N(j)$ we obtain,

$$\mathbb{E}\left[M_{rr}\right] = \mathbb{E}\left[\sum_{i \in S} \mathbb{1}_{i \in N(j)} \left(w_r^{(i)} \right)^2 \right] = \sum_{i \in S} \mathbb{E}\left[\mathbb{1}_{i \in N(j)} \left(w_r^{(i)} \right)^2 \right]$$

where $\mathbb{1}_{i \in N(j)}$ is an indicator random variable for the edge $\{i, j\}$ and takes the value 1 with probability p and 0 with probability $1 - p$. Therefore,

$$\mathbb{E}\left[M_{rr}\right] = \sum_{i \in S} \mathbb{E}\left[\mathbb{1}_{i \in N(j)} \left(w_r^{(i)} \right)^2 \right] = p \sum_{i \in S} \left(w_r^{(i)} \right)^2 = p \sum_{i \in S} \left(v_i^{(r)} \right)^2 = p \left\| \mathbf{v}^{(r)} \right\|^2 = p \,.$$

Similarly we obtain $\mathbb{E}\left[M_{rs}\right]$ for $r \neq s$ as

$$\begin{aligned} \mathbb{E}\left[M_{rs}\right] &= \mathbb{E}\left[\sum_{i \in N(j)} w_r^{(i)} w_s^{(i)} \right] = \mathbb{E}\left[\sum_{i \in S} \mathbb{1}_{i \in N(j)} w_r^{(i)} w_s^{(i)} \right] = \sum_{i \in S} \mathbb{E}\left[\mathbb{1}_{i \in N(j)} w_r^{(i)} w_s^{(i)} \right] \\ &= p \sum_{i \in S} w_r^{(i)} w_s^{(i)} = p \sum_{i \in S} v_i^{(r)} v_i^{(s)} = p \left\langle \mathbf{v}^{(r)}, \mathbf{v}^{(s)} \right\rangle = 0 \,. \end{aligned}$$

□

Acknowledgments: The work in this chapter is based on joint work with Akash Kumar and Anand Louis.

Chapter 4

Maximum Independent set in hypergraphs

Given a hypergraph $H = (V, E)$, the independent set problem asks to compute a set of vertices such that no hyperedge is completely contained inside the set. Finding the largest independent set problem is NP-hard, since the degenerate graph version of the problem is NP-hard [Kar72]. It follows from the work of [Hås97, Zuc07] that the problem is hard to approximate better than the factor of $n^{1-\varepsilon}$ for every $\varepsilon > 0$ unless P=NP.

Models and Results

Here, we study the problem in [FK01] semi-random model restated here for r-uniform hypergraphs as,

Definition 4.1 *Given parameters $n, k, r,$ and p, a hypergraph H is constructed as follows.*

1. *Let V be a set of n vertices. Fix an arbitrary subset $S \subset V$ of size k.*
2. *Add a hyperedge independently with probability p for each $r-$tuple of vertices $\{i_1, i_2, \ldots, i_r\}$, such that $\{i_1, i_2, \ldots, i_r\} \cap S \neq \{\}$ and $\{i_1, i_2, \ldots, i_r\} \cap (V \setminus S) \neq \{\}$. We denote the hypergraph induced by the collection of such r-tuples as $H[S, V \setminus S]$.*
3. *Allow a* monotone adversary *to add r-hyperedges arbitrarily to $H[S, V \setminus S]$ and hypergraph induced on $V \setminus S$ denoted by $H[V \setminus S]$.*

We introduce the following definition (from [KLP21, Kha21]) for notational convenience.

Definition 4.2 *Let* $f(r) \stackrel{\text{def}}{=} \dfrac{r^{5/2} 2^{3r-2} e^{3r/2-2}}{\sqrt{3}}$.

We relax the notion of recovery as in [MMT20] and allow a partial recovery by which we mean that we output an independent set of size $(1-\varepsilon)k$ for given $\varepsilon \in (0,1)$ which does not have to be the planted independent set. Formally we prove,

Theorem 4.1 *There exists a deterministic algorithm which takes as input $\varepsilon \in (0,1)$ and an instance of Definition 4.1 satisfying*

$$k \geq \max\left\{\frac{r2^{2r+2}e^r}{3p}, \frac{(2rf(r))^{1/(r-0.5)}n^{(r-1)/(r-0.5)}}{\varepsilon^{1/(r-0.5)}p^{1/(2r-1)}}\right\},$$

has running time $n^{\mathcal{O}(r)}$, and outputs an independent set of size at least $(1-\varepsilon)k$, with high probability (over the randomness of the input).

Related Works

The work [HL98b] gives a combinatorial algorithm for the maximum independent set problem to obtain an approximation ratio of $\mathcal{O}\left(n/\left(\log^{(r-1)} n\right)^2\right)$ for a r-uniform hypergraph where $\log^{(r)} n$ denotes a r-fold repeated application of logarithm as $\log \ldots \log n$. This has been improved by Halldórsson in the work [Hal00] where they study the problem on arbitrary weighted hypergraphs and give an $\mathcal{O}\left(n/\log n\right)$ approximation algorithm that runs in $\mathsf{poly}\left(n,m\right)$ time where m denotes the number of hyperedges. From here onwards, a lot of work has been done in studying the problem in special classes of graphs. In this section, we do a brief survey of these results.

The problem has been extensively studied for 3-uniform hypergraphs which contain an independent set of size γn. Krivelevich, Nathaniel, and Sudakov [KNS01] give an SDP-based algorithm that finds an independent set of size $\tilde{\Omega}\left(\min\left(n, n^{6\gamma-3}\right)\right)$ for $\gamma \geq 1/2$. The work by Chlamtac [Chl07] uses an SDP relaxation with the third level of the Lasserre/SoS hierarchy and returns an independent set of size $\Omega\left(n^{1/2-\gamma}\right)$. Chlamtac and Singh [CS08] gave an algorithm which computes an independent set of size $n^{\Omega(\gamma^2)}$ (where $\gamma \geq 0$ is a constant) using $\Theta(1/\gamma^2)$ levels of a mixed hierarchy which they called *the intermediate hierarchy*.

Halldórsson and Losievskaja [HL09] study the problem on bounded degree hypergraphs. For hypergraphs with degree bounded by Δ, the authors show that the classical greedy set cover algorithm can be analyzed to give $(\Delta+1)/2$ approximation. The work [KMM11a] shows that the bounded degree case is Unique Games-hard to approximate within a factor of $\mathcal{O}\left(\Delta/\log^2\Delta\right)$. In a recent work [BK19], the authors exhibit how to convert this inapproximability factor of $\mathcal{O}\left(\Delta/\log^2\Delta\right)$ under UG-hardness to NP-hardness.

The work [KLP21, Kha21] studies the maximum independent set problem in r-uniform

hypergraphs in [FK01] model. They consider a different relaxation of recover in the [FK01] model where they output a list of independent sets, one out of which is exactly the planted independent set w.h.p. They give recovery algorithms for $k = \Omega\left(n^{(r-1)/(r-0.5)}/p^{3/(2r-1)}\right)$. We note that our regimes of k, n, p in **??** is wider, however our notion of relaxation for recovery is different from theirs.

Lasserre/SOS Relaxation:

Theorem 4.1 generalizes to hypergraphs the analogous results for graphs by [MMT20]. Our proofs of these theorems are based on rounding "crude-SDP" in [MMT20] , augmented with "Lasserre/SoS like" hierarchy of constraints. Crude SDP is an idea developed by Kolla, Makarychev, and Makarychev, where the relaxation is written so that the vectors corresponding to the planted solution are clustered together. Crude-SDP has been used in the works [KMM11b, MMV12, MMT20]. The Lasserre/SoS hierarchy has been used in designing approximation algorithms for various problems as discussed in Chapter 2. The first step is to extend the crude-SDP in [MMT20] to r-uniform hypergraphs and we present our extension in SDP 4.1.

SDP 4.1

$$\max \sum_{\{i_1,i_2,\ldots,i_r\}\in\binom{V}{r}} \|x_{i_1,i_2,\ldots,i_r}\|^2$$

subject to

$$\|x_i\|^2 = 1 \qquad \forall i \in V \tag{4.1}$$

$$\|x_e\|^2 = 0 \qquad \forall e \in E \tag{4.2}$$

$$\langle x_I, x_J\rangle = \|x_{I\cup J}\|^2 \qquad \forall I, J(\neq\emptyset) \subseteq V,\ s.t\ |I\cup J| \leq r+1 \tag{4.3}$$

$$\langle x_u, x_I\rangle \geq \langle x_u, x_J\rangle \qquad \forall u \in V, \forall I \subseteq J \subseteq V, |J| \leq r+1 \tag{4.4}$$

$$1 - \|x_{u,v_1,\ldots,v_r}\|^2 \leq \sum_{i\in[r]} \left(1 - \|x_{u,v_i}\|^2\right) \qquad \forall \{u, v_1, \ldots, v_r\} \in \binom{V}{r+1}. \tag{4.5}$$

We let the optimal solution of the above SDP be denoted by $\{x^*_I\}_{I\subset V, 1\leq|I|\leq r+1}$. In [MMT20] they study a crude-SDP with the constraint $\langle x_i, x_j\rangle = 0, \forall \{i,j\} \in E$. Their crude SDP tries to cluster the vertices together, while the constraint $\langle x_i, x_j\rangle = 0, \{i,j\} \in E$ tries to ensure that no edges are contained in a cluster. Constraint 4.2 is a natural extension of this to hypergraphs. We add vectors for all subsets of vertices of size at most $r+1$, and add consistency constraints 4.3 among them, as in the Lasserre/SoS hierarchy. However, we note that SDP 4.1 is different from a Lasserre/SoS relaxation since there is no natural interpretation of solution to this crude-SDP

as a low-degree pseudo-distribution over independent sets in the hypergraph. However, we add the constraints in equation 4.3,4.4 and 4.5 since our intended feasible solution x' constructed as,

$$x'_{i_1,i_2,\ldots,i_l} = \begin{cases} \hat{e} & \text{if } \{i_1,i_2,\ldots,i_l\} \in \binom{S}{l} \\ x^*_{i_1,i_2,\ldots,i_l} & \text{if } \{i_1,i_2,\ldots,i_l\} \in \binom{V\setminus S}{l} \qquad \forall l \in [r+1] \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where $\hat{e}$ denote a unit vector orthogonal to x^*_I, $\forall I \subseteq V \setminus S$, $|I| \leq r$ and x^* is the optimal solution to the SDP. satisfies these constraints. The constraints in eq. (4.4) and eq. (4.5) are inspired from the locally consistent probability distributions viewpoint of a r-level Lasserre/SoS hierarchy [Rot13]. A t-level vector in a Lasserre/SoS hierarchy can be interpreted as the probability of the joint event corresponding to indices of the vector. Constraint 4.4 corresponds to the fact that the probability of a sub event can only be larger than the probability of an event and constraint 4.5 corresponds to a union bound on the complement of the joint event (represented by $x_{u,v_1,\ldots,v_r}$) given by sum of the complement of pairwise joint events $1 - x_{u,v_i}$ $\forall i \in [r]$.

4.1 SDP Bounding

Our analysis uses the SDP bounding idea where we prove a lower bound on the contribution of the SDP mass in the optimal solution from the r-level vectors of S, i.e. $\{x^*_I\}_{I\subset S,|I|=r}$. We use the following,

Lemma 4.1 *[KLP21, Kha21]*

$$\sum_{\{i_1 i_2 \ldots i_r\}\in\binom{S}{r}} \left\|x^*_{i_1,i_2,\ldots,i_r}\right\|^2 + \sum_{\{i_1,i_2,\ldots,i_r\}\in\partial(S)} \left\|x^*_{i_1,i_2,\ldots,i_r}\right\|^2 \geq \binom{k}{r}.$$

where the lower bound comes from the constructed solution in eq. (4.6). The key thing here is that the crude-SDP has allowed us to remove the contribution from $V \setminus S$ which corresponds to the part of graph that was fully controlled by adversary as defined in Definition 4.1. Similar to [MMT20], this just follows by comparing the constructed solution to the optimal solution and using the fact that the optimal will always have a larger objective value. We refer to the full version of the paper [KLP21] for the proof. However, some new ideas are required to bound the second term in hypergraphs.

As discussed, upper bounding the contribution from $S \times (V \setminus S)$, will give us a lower bound on the contribution from S. In [MMT20], $S \times (V \setminus S)$ is a random bipartite graph; they use Grothendieck's inequality and concentration bounds to upper bound the contribution from this

part. In our setting, $S \times (V \setminus S)$ is a random hypergraph, and [MMT20]'s techniques do not seem to be directly applicable here. The main idea is to construct a random bipartite graph $G' = (U_1, U_2, E')$ based on this random bipartite hypergraph as follows. One side of the graph consists of vertices corresponding to subsets of S of cardinality at most $r-1$, and the other side consists of vertices corresponding to subsets of $V \setminus S$ of cardinality at most $r-1$. We add an edge between two vertices if the union of the sets corresponding to them forms a hyperedge in our hypergraph. By this construction, $\sum_{\{a,b\} \in E'} \langle x_a, x_b \rangle$ is equal to the SDP mass from $S \times (V \setminus S)$ in our hypergraph. Moreover, since $S \times (V \setminus S)$ forms a random bipartite hypergraph, this construction gives us that G' is a random bipartite graph. Therefore, this can be used to bound the contribution from G' using [MMT20]'s approach (Proposition 4.1).

Proposition 4.1 *[KLP21, Kha21] For* $k \geq \dfrac{r 2^{2r+2} e^r}{3p}$,

$$\sum_{\{i_1, i_2, \ldots, i_r\} \in \partial(S)} \left\| x^*_{i_1, i_2, \ldots, i_r} \right\|^2 \leq \left(\frac{2^{3r-2} e^{3r/2-2}}{\sqrt{3} r^{r-5/2}} \right) \left(\sqrt{\frac{k}{p}} \right) n^{r-1} .$$

with high probability (over the randomness of the input).

We refer to the full version of the paper [KLP21] for a proof of this. Using lemma 4.1 we get that,

Corollary 4.1 *[KLP21, Kha21] For* $k \geq \dfrac{r 2^{2r+2} e^r}{3p}$,

$$\sum_{\{i_1, i_2, \ldots, i_r\} \in \binom{S}{r}} \left\| x^*_{i_1, i_2, \ldots, i_r} \right\|^2 \geq \binom{k}{r} - \left(\frac{2^{3r-2} e^{3r/2-2}}{\sqrt{3} r^{r-5/2}} \right) \left(\sqrt{\frac{k}{p}} \right) n^{r-1} .$$

with high probability (over the randomness of the input).

4.2 Algorithm for computing a large independent set

In this section, we will prove Theorem 4.1 which is a generalization of Theorem 1.1 of [MMT20] to r-uniform hypergraphs (Lemma 4.2, Lemma 4.3 and proof of Theorem 4.1). We will crucially use the lower bound on the SDP mass from the vectors in S, i.e., Corollary 4.1. As a first step towards this, in Lemma 4.2, we show that there exists a vertex $u \in S$ for which the 1 level vectors x^*_v (corresponding to vertices in S) in the optimal solution have a large projection on x^*_u.

Fact 4.1 (Bounds on Binomial Coefficient, Appendix C - [CLRS09]) *For $1 \leq k \leq n$,*

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k .$$

Lemma 4.2 *For $k \geq \dfrac{r2^{2r+2}e^r}{3p}$, there exists a vertex $u \in S$ such that, with high probability (over the randomness of the input).*

$$\mathbb{E}_{v \in S\setminus\{u\}} \left\langle x^*_u, x^*_v \right\rangle \geq \mathbb{E}_{\{i_1,i_2,\ldots,i_{r-1}\}\sim\binom{S\setminus\{u\}}{r-1}} \left\langle x^*_u, x^*_{i_1,i_2,\ldots,i_{r-1}} \right\rangle \geq 1 - \frac{f(r)n^{r-1}}{k^{r-0.5}\sqrt{p}} .$$

Proof: From Corollary 4.1 we have that for $k \geq \dfrac{r2^{2r+3}e^r}{3p}$,

$$\sum_{\{i_1,i_2,\ldots,i_r\}\in\binom{S}{r}} \left\| x^*_{i_1,i_2,\ldots,i_r} \right\|^2 \geq \binom{k}{r} - \frac{f(r)n^{r-1}\sqrt{k}}{r^r\sqrt{p}} .$$

From the SDP constraint 4.3, we split the above sum as follows,

$$\sum_{u\in S,\{i_1,i_2,\ldots,i_{r-1}\}\in\binom{S\setminus\{u\}}{r-1}} \left\langle x^*_u, x^*_{i_1,i_2,\ldots,i_{r-1}} \right\rangle \geq r\left(\binom{k}{r} - \frac{f(r)n^{r-1}\sqrt{k}}{r^r\sqrt{p}}\right) . \tag{4.7}$$

Therefore there exists a vertex $u \in S$ such that,

$$\sum_{\{i_1,i_2,\ldots,i_{r-1}\}\in\binom{S\setminus\{u\}}{r-1}} \left\langle x^*_u, x^*_{i_1,i_2,\ldots,i_{r-1}} \right\rangle \geq \frac{r}{k}\left(\binom{k}{r} - \frac{f(r)n^{r-1}\sqrt{k}}{r^r\sqrt{p}}\right) .$$

Since number of terms in expression in the above sum is $\binom{k-1}{r-1}$. We rewrite the above expression as an expectation over the uniform distribution on such tuples as,

$$\begin{aligned}
\mathbb{E}_{\{i_1,i_2,\ldots,i_{r-1}\}\sim\binom{S\setminus\{u\}}{r-1}} \left\langle x^*_u, x^*_{i_1,i_2,\ldots,i_{r-1}} \right\rangle &\geq \frac{r}{k\binom{k-1}{r-1}}\left(\binom{k}{r} - \frac{f(r)n^{r-1}\sqrt{k}}{r^r\sqrt{p}}\right) = 1 - \frac{rf(r)n^{r-1}\sqrt{k}}{k\binom{k-1}{r-1}r^r\sqrt{p}} \\
&\geq 1 - \frac{rf(r)n^{r-1}\sqrt{k}}{k\left(\dfrac{k-1}{r-1}\right)^{r-1}r^r\sqrt{p}} \geq 1 - \frac{rf(r)n^{r-1}\sqrt{k}}{k\left(\dfrac{k}{r}\right)^{r-1}r^r\sqrt{p}} \\
&= 1 - \frac{f(r)n^{r-1}}{k^{r-0.5}\sqrt{p}} .
\end{aligned}$$

where we used Fact 4.1 and the fact that, $(k-1)/(r-1) \geq k/r \iff k \geq r$.

Using our SDP constraint 4.4 we can rewrite the summation in eq. (4.7) as,

$$\sum_{u \in S, \{i_1,i_2,\ldots,i_{r-1}\} \in \binom{S \setminus \{u\}}{r-1}} \left\langle x_u^*, x_{i_1,i_2,\ldots,i_{r-1}}^* \right\rangle \leq \frac{1}{(r-1)} \sum_{u \in S, \{i_1,i_2,\ldots,i_{r-1}\} \in \binom{S \setminus \{u\}}{r-1}} \sum_{l=1}^{r-1} \left\langle x_u^*, x_{i_l}^* \right\rangle$$

$$= \frac{\binom{k-2}{r-2}}{(r-1)} \sum_{u \in S, v \in S \setminus \{u\}} \langle x_u^*, x_v^* \rangle \tag{4.8}$$

where the equality above can be argued by fixing a vertex $u \in S, v \in S \setminus \{u\}$ and observing that there are $\binom{k-2}{r-2}$ terms in the double summation containing such (u, v). We divide the eq. (4.8) by $k\binom{k-1}{r-1}$ (the number of terms in the summation on the left side) to rewrite the inequality in form of expectation as,

$$\mathbb{E}_{\{i_1,i_2,\ldots,i_{r-1}\} \sim \binom{S \setminus \{u\}}{r-1}} \left\langle x_u^*, x_{i_1,i_2,\ldots,i_{r-1}}^* \right\rangle \leq \frac{\binom{k-2}{r-2}}{(r-1)k\binom{k-1}{r-1}} \sum_{u \in S, v \in S \setminus \{u\}} \langle x_u^*, x_v^* \rangle$$

$$= \frac{1}{k(k-1)} \sum_{u \in S, v \in S \setminus \{u\}} \langle x_u^*, x_v^* \rangle = \mathbb{E}_{v \in S \setminus \{u\}} \langle x_u^*, x_v^* \rangle$$

where we have used the fact that $\binom{k-1}{r-1} = \frac{k-1}{r-1}\binom{k-2}{r-2}$. It then follows that there exists a vertex $u \in S$ such that

$$\mathbb{E}_{v \in S \setminus \{u\}} \langle x_u^*, x_v^* \rangle \geq 1 - \frac{f(r) n^{r-1}}{\sqrt{p} k^{r-0.5}}$$

□

Lemma 4.2 shows that a large fraction of the 1-level vectors in S have a large projection on x_u^*. We start with the following definition (similar to [KLP21, Kha21]),

Definition 4.3 *We denote the set of all l-tuples containing vertices from a set $T \subseteq V$ (where $l \leq |T|$) whose corresponding vectors have a projection at least $\mathcal{R}$ with the vector x_u^* by*

$$\mathcal{B}_u(l, \mathcal{R}, T) \stackrel{\text{def}}{=} \left\{ \{v_1, v_2, \ldots, v_l\} : \{v_1, v_2, \ldots, v_l\} \in \binom{T}{l} \text{ and } \left\langle x_u^*, x_{v_1,v_2,\ldots,v_l}^* \right\rangle \geq \mathcal{R} \right\}.$$

Note that the value of l of interest will be 1 in Theorem 4.1.

Lemma 4.3 *For $k \geq \frac{r2^{2r+2}e^r}{3p}$, there exists a vertex $u \in S$ such that*

$$\left|\mathcal{B}_u\left(1, 1-\frac{1}{2r}, S\right)\right| \geq (k-1)\left(1 - \frac{2rf(r)n^{r-1}}{\sqrt{p}k^{r-0.5}}\right)$$

with high probability (over the randomness of the input).

Proof: We note that $1 - \langle x_u^*, x_v^*\rangle \geq 0$ and for $\mathcal{R} \in (0,1)$ and for $k \geq \frac{r2^{2r+3}e^r}{3p}$, by applying Markov's inequality on $(1 - \langle x_u^*, x_v^*\rangle)$, where u is the vertex guaranteed in Lemma 4.2 and $v \in V \setminus S$ we have that,

$$\mathbb{P}_{v \in S\setminus\{u\}}\left[1 - \langle x_u^*, x_v^*\rangle > 1 - \mathcal{R}\right] < \frac{\frac{f(r)n^{r-1}}{\sqrt{p}k^{r-0.5}}}{1-\mathcal{R}}. \qquad \text{(using Lemma 4.2)}$$

We can rewrite the above expression as the fraction of vertices which satisfy $(\langle x_u^*, x_v^*\rangle < \mathcal{R})$, since the underlying distribution is the uniform distribution over all such v and by setting $\mathcal{R} = 1 - 1/2r$,

$$\left|v \in S \setminus \{u\} : \langle x_u^*, x_v^*\rangle < 1 - \frac{1}{2r}\right| < (k-1)\left(\frac{2rf(r)n^{r-1}}{\sqrt{p}k^{r-0.5}}\right).$$

$$\therefore \left|\mathcal{B}_u\left(1, 1-\frac{1}{2r}, S\right)\right| = \left|v \in S \setminus \{u\} : \langle x_u^*, x_v^*\rangle \geq 1 - \frac{1}{2r}\right| \geq (k-1)\left(1 - \frac{2rf(r)n^{r-1}}{\sqrt{p}k^{r-0.5}}\right).$$

□

In [MMT20] they use the SDP constraint $\langle x_u, x_v\rangle = 0, \forall \{u,v\} \in E$ to show that the set of vectors which have a large projection on x_u^* is an independent set. Therefore they directly analyze the bound on the size of the set to obtain an independent set, in a range of p such that it covers at least $(1-\varepsilon)$ fraction of vertices in S. However, in our setting, we are unable to guarantee directly that this set of vectors is an independent set. We crucially use the Lasserre/SoS like SDP constraints 4.3 and 4.5 and an appropriately large value of $\mathcal{R}$ $(\mathcal{R} \geq 1 - \frac{1}{2r})$ to show that the set guaranteed in Lemma 4.3 is an independent set.

Lemma 4.4 *For $k \geq \frac{r2^{2r+2}e^r}{3p}$, there exists a vertex $u \in S$ such that $\mathcal{B}_u\left(1, 1-\frac{1}{2r}, V\right)$ is an independent set with high probability (over the randomness of the input).*

Proof: We consider the SDP constraint 4.5 and apply it to our optimal solution x^* . By using consistency constraints $(\langle x_I, x_J\rangle = \langle x_{I'}.x_{J'}\rangle, \forall I \cup J = I' \cup J')$ (eq. (4.3)) we can rewrite

the constraint in Equation (4.5) as,

$$1 - \left\|x^*_{u,i_1,\ldots,i_r}\right\|^2 \leq \sum_{l\in[r]} \left(1 - \left\langle x^*_u, x^*_{i_l}\right\rangle\right) . \tag{4.9}$$

For $k \geq \dfrac{r2^{2r+3}e^r}{3p}$, if we pick any set of r vertices $\{i_1,\ldots,i_r\} \in \binom{V}{r}$ in $\mathcal{B}_u\left(1, 1-\dfrac{1}{2r}, V\right)$ (where u is the vertex guaranteed in Lemma 4.3) we know that $\left\langle x^*_u, x^*_{i_l}\right\rangle \geq 1 - \dfrac{1}{2r}, \forall l \in [r]$. By using eq. (4.9) we have that,

$$\left\|x^*_{u,i_1,\ldots,i_r}\right\|^2 \geq 1 - \sum_{l\in[r]} \left(1 - \left\langle x^*_u, x^*_{i_l}\right\rangle\right) \geq 1 - \sum_{l\in[r]} \frac{1}{2r} \geq \frac{1}{2} > 0\,. \tag{4.10}$$

Now we examine the term $\left\|x^*_{i_1,i_2,\ldots,i_r}\right\|^2$ for these $\{i_l,\ldots,i_r\}$ and we have that,

$$\left\|x^*_{i_1,i_2,\ldots,i_r}\right\|^2 = \left\langle x^*_{i_1}, x^*_{i_2\ldots,i_r}\right\rangle \geq \left\langle x^*_{i_1}, x^*_{u,i_2\ldots,i_r}\right\rangle = \left\|x^*_{u,i_1,\ldots,i_r}\right\|^2 > 0$$

where the equality holds by consistency constraints, the first inequality above holds by constraint 4.4 and the last inequality holds by eq. (4.10). Hence for any r-tuple $\{i_1,i_2,\ldots,i_r\} \subseteq \mathcal{B}_u\left(1, 1-\dfrac{1}{2r}, V\right)$, we have $\left\|x^*_{i_1,i_2,\ldots,i_r}\right\|^2 > 0$. Therefore by SDP constraint 4.2, it cannot form a hyperedge. Hence, the set of vertices in $\mathcal{B}_u\left(1, 1-\dfrac{1}{2r}, V\right)$ is an independent set. □

We introduce a definition from [KLP21, Kha21],

Definition 4.4 *Let $\mathcal{S}_u$ denote the set of vertices formed by the union of all vertices by reading off the indices from the tuples of the set, $\mathcal{B}_u(l,r,V)$.*

Now, we have all the ingredients to prove our main result. We present the complete algorithm below (similar to [KLP21, Kha21]) and the proof of Theorem 4.1.

Algorithm 3:

Require: $H = (V, E)$, $l \in [r]$, and $\mathcal{R} \in (0, 1)$.

Ensure: A list of independent sets in H.

1: Solve SDP 4.1.
2: **for all** $u \in V$ **do**
3: Initialize $\mathcal{S}_u$ denote the union of set of vertices from the tuples in $\mathcal{B}_u(l, \mathcal{R}, V)$.
4: $\mathcal{S}'_u = \{u\} \cup \mathcal{S}_u$. If $\mathcal{S}'_u$ is not an independent set,
Set $\mathcal{S}'_u = \emptyset$ and skip this iteration.
5: **for all** $v \in V \setminus \mathcal{S}_u$ **do**
6: Add vertex v to $\mathcal{S}'_u$ if $\mathcal{S}'_u \cup \{v\}$ is an independent set.
7: **end for**
8: **end for**
9: Return $\{\mathcal{S}'_u\}_{u \in V}$.

We set our parameters (n, p, k, ε) appropriately and show that the number of vertices in $\mathcal{B}_u$ along with the vertex u (denoted by $\mathcal{S}'_u$) cover $1 - \varepsilon$ fraction of vertices in S.

Proof: [Proof of Theorem 4.1] We run the Algorithm 3 with the inputs, $H, l = 1$ and $\mathcal{R} = 1 - \frac{1}{2r}$ to get $\{\mathcal{S}'_u\}_{u \in V}$. In Lemma 4.3 we show that

$$\left|\mathcal{B}_u\left(1, 1 - \frac{1}{2r}, S\right)\right| \geq (k-1)\left(1 - \frac{2rf(r)n^{r-1}}{\sqrt{p}k^{r-0.5}}\right).$$

For a suitable choice of parameters we wish to have,

$$\left|\mathcal{B}_u\left(1, 1 - \frac{1}{2r}, S\right)\right| \geq (k-1)(1-\varepsilon). \tag{4.11}$$

We can then include in the vertex u to our independent set and we get

$$\begin{aligned}|\mathcal{S}'_u| \geq |\mathcal{S}_u| + 1 = \left|\mathcal{B}_u\left(1, 1 - \frac{1}{2r}, V\right)\right| + 1 \geq \left|\mathcal{B}_u\left(1, 1 - \frac{1}{2r}, S\right)\right| + 1 \\ \geq (k-1)(1-\varepsilon) + 1 \geq k(1-\varepsilon).\end{aligned}$$

We note that by setting $k \geq \frac{(2rf(r))^{1/(r-0.5)} n^{(r-1)/(r-0.5)}}{\varepsilon^{1/(r-0.5)} p^{1/(2r-1)}}$, equation 4.11 is satisfied and hence we can recover an independent set of size $(1-\varepsilon)k$ for all $\varepsilon \in (0, 1)$. □

Acknowledgments: The work in this chapter is based on joint work with Yash Khanna and Anand Louis [KLP21]. Theorem 1.2 of [KLP21] appears in [Kha21] and Theorem 1.3 of this

work is presented here (Theorem 4.1).

Chapter 5

Largest Induced Planar Subgraph

In this chapter, we study the problem of finding the largest induced planar subgraph of a given graph $G = (V, E)$. The model is constructed as per the *planted solution model* (Definition 1.1).

Problem 5.1 *Given a graph $G = (V, E)$, the largest induced planar subgraph problem asks us to find a maximal set $S \subseteq V$ such that the subgraph induced on S is planar.*

Related Works

The worst-case analysis gives us that the largest induced planar subgraph problem is NP-hard owing to the hereditary structure of planar graphs (follows from [Yan78]). The work by [LY93] shows that there exists an $\varepsilon > 0$ such that this problem cannot be approximated with ratio n^{ε} in polynomial time unless P=NP. The work [Hal00] gives $\mathcal{O}\left(n^{-1}\left(\log n/\log\log n\right)^2\right)$ approximation algorithm for the problem.

The work [HL98a] studies the problem for graphs with degree bounded by d and gives a linear time algorithm with approximation ratio $1/\lceil(d+1)/3\rceil$. This was improved[1] in the work [EF02] which gives a $3/(d+1)$ approximation algorithm that runs in time $\mathcal{O}(mn)$.

The problem is equivalent to the *nonplanar vertex deletion* problem where the task is to remove minimum number of vertices whose removal leaves a planar graph. The work [FdFG+06] shows that the problem is NP-hard even when restricted to cubic graphs. They also show that there exists no constant factor approximation algorithm unless P=NP. The work [RS95] gives a fixed parameter tractable algorithm with running time $\mathcal{O}(f(k)n^3)$ where the parameter k is the number of vertices required to be deleted. The work [KS17a] gives a $\mathcal{O}\left(\log^{32} n\right)$-approximation algorithm with running time $n^{\mathcal{O}(\log n/(\log\log n))}$ and an order $\mathcal{O}(n^{\varepsilon})$-approximation algorithm running in time $n^{\mathcal{O}(1/\varepsilon)}$ for any constant $\varepsilon > 0$.

[1] Improvement for the case when $d \mod 3 \neq 2$.

A related problem is the *nonplanar edge deletion* which is equivalent to finding maximum planar subgraph. This problem was also shown to be NP-hard in the work [LG79] but the work [CFFK98] shows that it admits a 4/9 approximation algorithm.

Our Results

In this chapter we study Problem 5.1 in the planted solution model (Definition 1.1). We now present our main result in this model.

Theorem 5.1 *For* $k \geq \frac{224\sqrt{n}}{p^2}$ *where* $p = \Omega\left(\frac{\log k}{k}\right)$, *there exists a deterministic algorithm which given an instance generated as per Definition 1.1, returns a list of sets (of size* $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$*) such that atleast one set in the list is the planted set* S *with high probability (over the randomness of the input) in time* $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$

Proof Overview

We let A denote the adjacency matrix of the graph constructed for this problem as per the planted solution model. We let $B = \bar{A}$ denote the adjacency matrix of the complement graph so that $A + B = J - I$. We consider an equivalent way of generating the graph corresponding to matrix B (equivalent to the planted solution model) by the following series of steps,

1. Start with a random graph $\mathcal{G}_{n,1-p}$ and let the matrix corresponding to the graph so far be denoted by B_R.

2. Add edges in the planted region $S \times S$ such that we obtain a complete graph on S. The distribution of the graph corresponding to edges added in this step is $\mathcal{G}_{k,p}$ and we denote the matrix corresponding to this graph as B_C.

3. Remove the edges corresponding to the edges in the planar graph $A_{S\times S}$. These are at most $3k - 6$ edges, and we denote the associated matrix by B_W.

Hence, we can express the resulting matrix for the complement graph as

$$B = B_R + B_C + B_W\,. \tag{5.1}$$

The B_W term is the only term that is different from the matrix corresponding to the planted clique problem, and in Lemma 5.1 we show that $\|B_W\| = \mathcal{O}\left(\sqrt{k}\right)$. In the rest of Section 5.1, we reproduce the calculations for the planted clique problem as in the work [AKS98, Tre17], and recover $(1-\delta)$ fraction of vertices (we denote these set of vertices as T) of the planted set S

for given $\delta > 0$. In Section 5.2 we do a post-processing step, by computing the degree of rest of the vertices to the set T. Using this information, we recover all the low degree vertices (vertices with degree $\leq pk/8$) of S. These remaining high vertices are recovered by enumerating over all possible sets of size $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$.

5.1 Partial recovery of planted planar graph

The B_W term in eq. (5.1) is the only term that is different from the matrix corresponding to the planted clique problem. In order to use the analysis for the planted clique problem from the works [AKS98, Tre17], we start by bounding the spectral norm of B_W in Lemma 5.1.

Lemma 5.1 $\|B_W\|_2 \leq \sqrt{6k}$.

Proof: Since the planted graph is planar, we know that it cannot have more than $3k - 6$ edges. We then have that,

$$\|A_{S\times S}\|_2 \leq \|A_{S\times S}\|_F \leq \sqrt{6k-12} \leq \sqrt{6k}\,.$$

Now we are done simply by noting the way we have defined B_W, i.e., $B_W = A_{S\times S}$. □

Lemma 5.2 *For an instance of planted planar graph given by Definition 1.1 and a parameter $\delta > 0$ in regimes where $k \geq (28\sqrt{n})/p\delta$, there exists a deterministic algorithm which can recover at least $(1-\delta)$ fraction of the planted vertices in S.*

Proof: We start by considering the matrix $B - (1-p)J$ and using Fact 2.1, we note that,

$$\|B-(1-p)J\|_2 \geq \frac{\mathbb{1}_S^T B \mathbb{1}_S}{\mathbb{1}_S^T \mathbb{1}_S} - (1-p)\frac{\mathbb{1}_S^T J \mathbb{1}_S}{\mathbb{1}_S^T \mathbb{1}_S} \geq \frac{k^2 - k - 2(3k-6) - (1-p)k^2}{k} \geq pk - 7\,.$$

Using Claim 2.1 we have that almost surely,

$$\|B_R - (1-p)J\| \leq 2\sqrt{n} \quad \text{and} \quad \left\|B_C - p\mathbb{1}_S\mathbb{1}_S^T\right\| \leq 2\sqrt{k}\,.$$

Using Lemma 5.1 we have that,

$$\|B_W\| \leq \sqrt{6k} \leq 3\sqrt{k}\,.$$

Now let $\mathbf{x}$ be the eigenvector corresponding to largest eigenvalue of $B - (1-p)J$ and putting

everything together and setting $\|\mathbf{x}\| = 1$ we obtain that,

$$\mathbf{x}^T(B - (1-p)J)\mathbf{x} = \mathbf{x}^T\left(B_R - (1-p)J + B_C - p\mathbb{1}_S\mathbb{1}_S^T + p\mathbb{1}_S\mathbb{1}_S^T + B_W\right)\mathbf{x}\,.$$

Therefore we have that,

$$p\mathbf{x}^T\left(\mathbb{1}_S\mathbb{1}_S^T\right)\mathbf{x} \geq pk - 7 - 2\sqrt{n} - 5\sqrt{k} \geq pk\left(1 - \frac{\delta}{2}\right)$$

where the last inequality follows form our choice of k. Now taking square root on both sides we obtain that,

$$\langle \mathbf{x}, \mathbb{1}_S\rangle \geq \sqrt{k}\sqrt{1-\frac{\delta}{2}} \geq \sqrt{k}\left(1 - \frac{\delta}{4}\right) \text{ and hence } \left\|\sqrt{k}\mathbf{x} - \mathbb{1}_S\right\|^2 \leq 2k - 2\sqrt{k}\,\langle \mathbf{x}, \mathbb{1}_S\rangle \leq \frac{\delta k}{2}\,. \tag{5.2}$$

Let T be a set of k vertices formed by taking the absolute value of entries in the eigenvector $\mathbf{x}$ and picking top k entries. We let t be the smallest entry in the set T and $B \stackrel{\text{def}}{=} S \setminus T$. Scaling $\mathbf{x}$ by $\sqrt{k}$ to compare with $\mathbb{1}_S$, and using $|x_i| \geq t/\sqrt{k}, \forall i \in T$ we have that,

$$\begin{aligned}\left\|\sqrt{k}\mathbf{x} - \mathbb{1}_S\right\|^2 &= \sum_{i\in S}\left(\sqrt{k}x_i - 1\right)^2 + \sum_{i\notin S} kx_i^2 \geq \sum_{i\in S\setminus T}\left(\sqrt{k}x_i - 1\right)^2 + \sum_{i\in T\setminus S} kx_i^2 \\ &\geq |B|\,(1-t)^2 + |B|\,t^2 \geq \frac{|B|}{2} \quad \left(\text{Using } \min\left\{(1-t)^2, (1+t)^2\right\} = (1-t)^2\right)\,.\end{aligned} \tag{5.3}$$

where the last inequality follows from optimizing $f(t) = (1-t)^2 + t^2$ and noting that minimum occurs for $t = 1/2$. Therefore, using eq. (5.2) and eq. (5.3) we have that $|B| \leq \delta k$ and we recover $(1-\delta)$ fraction of vertices in planted set S. □

5.2 Full recovery of planted planar graph

Let the set $\mathcal{S} \subseteq S$ be the set of vertices in S which have low degree (degree $\leq pk/8$). As a first step towards full recovery, we aim to recover this set $\mathcal{S}$. We will identify the vertices in set $\mathcal{S}$ by their degree to the set T.

Since the number of edges in S is $3k-6$, the average degree of vertices in S is ≤ 6. Expressing the average as an expectation over uniform distribution over vertices in S and using Markov's

inequality we have that,

$$\mathbb{P}\left[\mathsf{deg}(v,S) \geq \frac{pk}{8}\right] \leq \frac{48}{pk} \text{ and therefore } \left|v : \mathsf{deg}(v,S) \geq \frac{pk}{8}\right| \leq \frac{48}{p}\,.$$

Therefore $|S \setminus \mathcal{S}| \leq 48/p$. We recall our aim is to recover the vertices in set $\mathcal{S}$, however, in Lemma 5.3 we recover a set $\mathcal{S}'$ which is a superset of $\mathcal{S}$ (may also contain vertices in the set $S \setminus \mathcal{S}$ in addition to the set $\mathcal{S}$). We do so by considering the degree of an vertex v to the set T denoted by $\mathsf{deg}(v,T)$. However, T is not a fixed set but a function of the randomness of the input instance. Therefore, computing the degree to set T doesn't seem easy. Hence we compute the degree of an arbitrary vertex to the set S instead.

Lemma 5.3 *Given a set T of size k such that $|S \cap T| \geq (1-\delta)\,k$ for any $\delta > 0$, in the regimes of $k \geq \dfrac{224\sqrt{n}}{p^2}$ and $p = \Omega\left(\dfrac{\log k}{k}\right)$ Algorithm 4 outputs a list of sets (of size $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$), one out of which is the planted set S with high probability (over the randomness of the input).*

Proof: For a vertex v in $V \setminus S$ we have that,

$$\mathbb{E}\left[\mathsf{deg}(v,S)\right] = pk\,.$$

where the expectation is over the randomness of the input instance. Using Chernoff bounds (Fact 2.3) we have that for a fixed vertex $v \in V \setminus S$,

$$\mathbb{P}\left[\mathsf{deg}(v,S) \leq \frac{pk}{2}\right] \leq \exp\left(-\frac{pk}{4}\right)\,.$$

We do a union bound over all vertices $v \in V \setminus S$ and we get that,

$$\mathbb{P}\left[\exists v \in V \setminus S : \mathsf{deg}(v,S) \leq \frac{pk}{2}\right] \leq n\exp\left(-\frac{pk}{4}\right)\,.$$

Therefore with high probability (over the randomness of the input instance) and for $p = \Omega\left(\log k/k\right)$, for any vertex $v \in V \setminus S$ we have that $\mathsf{deg}(v,S) \geq pk/2$.

$$\mathsf{deg}(v,T) \geq \mathsf{deg}(v,S) - |S \setminus T| \geq \frac{pk}{2} - \delta k\,.$$

While for a vertex $v \in \mathcal{S}$ we have,

$$\mathsf{deg}(v,T) \leq \mathsf{deg}(v,S) + |S \setminus T| \leq \frac{pk}{8} + \delta k\,.$$

Therefore, we can distinguish the set to which a vertex belongs if,

$$\frac{pk}{8} + \delta k < \frac{pk}{2} - \delta k \text{ which holds if } \delta k \leq \frac{pk}{8}\,.$$

For the remaining $\mathcal{O}(1/p)$ set of vertices which belong to the set $S \setminus \mathcal{S}'$, we simply enumerate over all possible sets of vertices of size $\leq 48/p$ and add to the set $\mathcal{S}'$ if it forms a planar graph. Therefore, we return a list of planar graphs, one out of which is the planted planar graph with high probability (over the randomness of the input). The running time of the algorithm therefore is $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$. □

Proof: [Proof of Theorem 5.1] Given an instance of a planted planar graph as per Definition 1.1, for $k \geq (224\sqrt{n})/p^2$, using Lemma 5.2 we can recover $(1-\delta)k$ vertices for $\delta \leq p/8$. Using guarantees of Lemma 5.3 we can then output a list $\mathcal{L}$ of size $\mathcal{O}\left(n^{\mathcal{O}(1/p)}\right)$ having planar graphs, one out of which is the planted planar graph S. □

Next, we present our algorithm based on guarantees of Theorem 5.1.

Algorithm 4:

Require: Given $G = (V, E)$, p, k, n.

Ensure: A list of planar graphs.

1: Initialize $\mathcal{L} = \phi$.
2: Compute the largest eigenvector $\mathbf{x}$ of $B - (1-p)J$.
3: Let T be the set of vertices corresponding to top k entries in the eigenvector $\mathbf{x}$ when sorted according to their absolute values.
4: Let $\mathcal{S}'$ be the set of vertices which have their degree to T, $\mathsf{deg}(v, T) \leq pk/4$.
5: **for all** $S' \subseteq V : |S'| \leq 48/p$ **do**
6: **if** $S' \cup \mathcal{S}'$ induces a planar subgraph **then**
7: $\mathcal{L} = \mathcal{L} \cup \{S' \cup \mathcal{S}'\}$.
8: **end if**
9: **end for**

Acknowledgments: The work in this chapter is based on joint work with Akash Kumar and Anand Louis.

Chapter 6

Conclusion

In this thesis, we study three graph problems, where the task is to find the largest induced subgraph with some structure, namely the Largest induced planar subgraph problem, the Odd Cycle Transversal problem, and the maximum independent set in hypergraph problem. The worst-case instances of these problems are intractable, and hence we study them in various random and semi-random models in Chapter 5, Chapter 3 and Chapter 4 respectively. We conclude with a few open problems.

1. For the largest induced bipartite subgraph problem, we give an algorithm that works for $k = \Omega_p\left(\sqrt{n \log n}\right)$ for a large enough range of p (includes regimes when $p = o(1)$). Achieving exact recovery for $k = \Omega\left(\sqrt{n}\right)$ in $p = o(1)$ regimes is still an open problem, to the best of our knowledge. We believe our SDP 3.1 is integral even for $k = \Omega_p\left(\sqrt{n}\right)$ and leave the task of proving it as an open problem.

2. The [MMT20] style relaxation of [FK01] model where the algorithm is allowed to output a list of n independent sets, that includes the planted independent set w.h.p is more natural and arguably a weaker relaxation. Whether one can extend $k = \Omega_p\left(n^{2/3}\right)$ result of [MMT20] in the largest induced bipartite subgraph problem is another interesting question, unanswered to the best of our knowledge.

3. More generally, achieving recovery for $k = o\left(n^{2/3}\right)$ in [MMT20] style relaxation of [FK01] model would imply an improvement for the planted clique/independent set problem. Therefore, one could ask the same question for the special case of planted independent set problem i.e. whether the gap between $k = \Omega_p\left(n^{2/3}\right)$ and $k = \Omega_p\left(\sqrt{n}\right)$ (exact recovery algorithms are known in the weaker *sandwich model*) can be closed. We refer to Chapter 9 in the book [Rou21] for an elaborate discussion on this.

4. For the problem of Maximum Independent sets in hypergraphs, one could ask whether exact recovery is possible for $k = \Omega_p(\sqrt{n})$ in the weaker (in strength compared to [FK01] model) but adversarial *sandwich model*. The degenerate graph version of the problem has been solved by the work [FK00]. However, to the best of our knowledge, the problem in hypergraphs is still open.

Bibliography

[ABBS14] Emmanuel Abbe, Afonso S. Bandeira, Annina Bracher, and Amit Singer. Decoding binary node labels from censored edge measurements: phase transition and efficient recovery. *IEEE Trans. Network Sci. Eng.*, 1(1):10–22, 2014. 17, 26

[ABH16] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Trans. Inform. Theory*, 62(1):471–487, 2016. 5, 6, 17, 26

[ACMM05] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 573–581. ACM, New York, 2005. 21

[AG11] Sanjeev Arora and Rong Ge. New tools for graph coloring. In *Approximation, randomization, and combinatorial optimization*, volume 6845 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, Heidelberg, 2011. 24, 27

[AK97] Noga Alon and Nabil Kahalé. A spectral technique for coloring random 3-colorable graphs. *SIAM J. Comput.*, 26(6):1733–1748, 1997. 5, 6

[AKRR90] A. Agrawal, P. Klein, S. Rao, and R. Ravi. Approximation through multicommodity flow. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 726–737 vol.2, Los Alamitos, CA, USA, oct 1990. IEEE Computer Society. 21

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. In *Proceedings of the Eighth International Conference "Random Structures and Algorithms" (Poznan, 1997)*, volume 13, pages 457–466, 1998. 2, 3, 22, 23, 64, 65

[AM99] Claudio Arbib and Raffaele Mosca. Polynomial algorithms for special cases of the balanced complete bipartite subgraph problem. *JCMCC. The Journal of Combinatorial Mathematics and Combinatorial Computing*, 30, 01 1999. 18

[AP89] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Appl. Math.*, 23(1):11–24, 1989. 1

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222–231. ACM, New York, 2004. 17

[BCC+10] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities—an $O(n^{1/4})$ approximation for densest k-subgraph. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 201–210. ACM, New York, 2010. 6, 17

[BHK+16] Boaz Barak, Samuel B. Hopkins, Jonathan Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*, pages 428–437. IEEE Computer Soc., Los Alamitos, CA, 2016. 8, 22

[BK09] Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2009*, pages 453–462. IEEE Computer Soc., Los Alamitos, CA, 2009. 21

[BK19] Amey Bhangale and Subhash Khot. UG-Hardness to NP-Hardness by Losing Half. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 53

[BL12] Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combin. Probab. Comput.*, 21(5):643–660, 2012. 1

[Bod88] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Automata, languages and programming (Tampere, 1988)*, volume 317 of *Lecture Notes in Comput. Sci.*, pages 105–118. Springer, Berlin, 1988. 1

[Bop87] Ravi B. Boppana. Eigenvalues and graph bisection: An average-case analysis (extended abstract). In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 280–285. IEEE Computer Society, 1987. 6

[BS95] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *J. Algorithms*, 19(2):204–234, 1995. 3, 5

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004. 9, 16, 39

[CC00] Yizong Cheng and George M. Church. Biclustering of expression data. In Philip E. Bourne, Michael Gribskov, Russ B. Altman, Nancy Jensen, Debra A. Hope, Thomas Lengauer, Julie C. Mitchell, Eric D. Scheeff, Chris Smith, Shawn Strande, and Helge Weissig, editors, *ISMB*, pages 93–103. AAAI, 2000. 18

[CFFK98] Gruia Călinescu, Cristina G. Fernandes, Ulrich Finkler, and Howard Karloff. A better approximation algorithm for finding planar subgraphs. volume 27, pages 269–302. 1998. 7th Annual ACM-SIAM Symposium on Discrete Algorithms (Atlanta, GA, 1996). 64

[Chl07] E. Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 691–701, 2007. 17, 53

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009. 57

[CO07] Amin Coja-Oghlan. Colouring semirandom graphs. *Combin. Probab. Comput.*, 16(4):515–552, 2007. 6, 17, 26

[CS08] Eden Chlamtac and Gyanit Singh. Improved approximation guarantees through higher levels of SDP hierarchies. In *Approximation, randomization and combinatorial optimization*, volume 5171 of *Lecture Notes in Comput. Sci.*, pages 49–62. Springer, Berlin, 2008. 17, 53

[CX16] Yudong Chen and Jiaming Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *J. Mach. Learn. Res.*, 17:Paper No. 27, 57, 2016. 22, 33

[CZ20] Sam Cole and Yizhe Zhu. Exact recovery in the hypergraph stochastic block model: a spectral algorithm. *Linear Algebra Appl.*, 593:45–73, 2020. 6

[DF16] Roee David and Uriel Feige. On the effect of randomness on planted 3-coloring models. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 77–90. ACM, New York, 2016. 6

[DK70] Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM J. Numer. Anal.*, 7:1–46, 1970. 11

[EF02] Keith Edwards and Graham Farr. An algorithm for finding large induced planar subgraphs. In *Graph drawing (Vienna, 2001)*, volume 2265 of *Lecture Notes in Comput. Sci.*, pages 75–83. Springer, Berlin, 2002. 63

[Epp94] David Eppstein. Arboricity and bipartite subgraph listing algorithms. *Inform. Process. Lett.*, 51(4):207–211, 1994. 22

[FdFG+06] Luerbio Faria, Celina M. Herrera de Figueiredo, Sylvain Gravier, Candido F. X. de Mendonça, and Jorge Stolfi. On maximum planar induced subgraphs. *Discrete Appl. Math.*, 154(13):1774–1782, 2006. 63

[FG95] U. Feige and M. Goemans. Approximating the value of two power proof systems, with applications to max 2sat and max dicut. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 182–189, 1995. 17

[FGR+13] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S. Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *STOC'13—Proceedings of the 2013 ACM Symposium on Theory of Computing*, pages 655–664. ACM, New York, 2013. 8, 22

[FK00] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures Algorithms*, 16(2):195–208, 2000. 4, 17, 22, 26, 70

[FK01] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. volume 63, pages 639–671. 2001. Special issue on FOCS 98 (Palo Alto, CA). iii, vii, 4, 5, 6, 7, 8, 17, 18, 19, 21, 26, 48, 52, 54, 69, 70

[FK04] Uriel Feige and Shimon Kogan. Hardness of approximation of the balanced complete bipartite subgraph problem. Technical report, 2004. 21

[FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Found. Trends Theor. Comput. Sci.*, 14(1-2):front matter, 1–221, 2019. 9

[FLS+18] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, MichałPilipczuk, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *ACM Trans. Algorithms*, 14(3):Art. 34, 45, 2018. 1

[FR10] Uriel Feige and Dorit Ron. Finding hidden cliques in linear time. In *21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA'10)*, Discrete Math. Theor. Comput. Sci. Proc., AM, pages 189–203. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2010. 22

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractability*. A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness. 21

[GL21] Suprovat Ghoshal and Anand Louis. *Approximation Algorithms and Hardness for Strong Unique Games*, pages 414–433. 01 2021. 21

[GLR19] Suprovat Ghoshal, Anand Louis, and Rahul Raychaudhury. Approximation algorithms for partially colorable graphs. In *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, volume 145 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 28, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. 21, 24, 27, 32

[GM75] G. R. Grimmett and C. J. H. McDiarmid. On colouring random graphs. *Math. Proc. Cambridge Philos. Soc.*, 77:313–324, 1975. 2

[GVY98] Naveen Garg, Vijay Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, 25, 01 1998. 21

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995. 17

[Hal00] Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.*, 4:no. 1, 16, 2000. 53, 63

[Hås97] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Electron. Colloquium Comput. Complex.*, 4(38), 1997. 52

[HKP+17] Samuel B. Hopkins, Pravesh K. Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures, 2017. 17

[HL98a] Magnús Halldórsson and Hoong Lau. Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and coloring. *Journal of Graph Algorithms and Applications*, 1, 04 1998. 63

[HL98b] Thomas Hofmeister and Hanno Lefmann. Approximating maximum independent sets in uniform hypergraphs. In *Mathematical foundations of computer science, 1998 (Brno)*, volume 1450 of *Lecture Notes in Comput. Sci.*, pages 562–570. Springer, Berlin, 1998. 53

[HL09] Magnús M. Halldórsson and Elena Losievskaja. Independent sets in bounded-degree hypergraphs. *Discrete Appl. Math.*, 157(8):1773–1786, 2009. 53

[HLL83] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137, 1983. 6

[HSS15] Samuel B. Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-squares proofs, 2015. 17

[HSSS16] Samuel B. Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors, 2016. 17

[HWX16a] Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Trans. Inform. Theory*, 62(5):2788–2797, 2016. 6

[HWX16b] Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming: extensions. *IEEE Trans. Inform. Theory*, 62(10):5918–5937, 2016. 6

[HWX16c] Bruce Hajek, Yihong Wu, and Jiaming Xu. Semidefinite programs for exact recovery of a hidden community, 2016. 6

[Joh87] David S Johnson. The np-completeness column: An ongoing guide. *Journal of Algorithms*, 8(3):438–448, 1987. 21

[Kar72] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972. 52

[Kha21] Yash Khanna. Robust algorithms for recovering planted structures in semi-random instances. Master's thesis, Indian Institute of Science, 4 2021. 52, 53, 55, 56, 58, 60, 61

[KL20] Yash Khanna and Anand Louis. Planted models for the densest k-subgraph problem. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 182 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. 27, 18. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020. 7

[KLP21] Yash Khanna, Anand Louis, and Rameesh Paul. Independent sets in semi-random hypergraphs. *CoRR*, abs/2104.00927, 2021. 6, 52, 53, 55, 56, 58, 60, 61

[KLT17] Akash Kumar, Anand Louis, and Madhur Tulsiani. Finding pseudorandom colorings of pseudorandom graphs. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 93 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 37, 12. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017. 6, 24, 27

[KMM11a] Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique games against a semi-random adversary: Study of semi-random models of unique games. *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, Oct 2011. 53

[KMM11b] Alexandra Kolla, Konstantin Makarychev, and Yury Makarychev. How to play unique games against a semi-random adversary: study of semi-random models of unique games. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*, pages 443–452. IEEE Computer Soc., Los Alamitos, CA, 2011. 54

[KMS98] David Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. 17

[KNS01] Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *J. Algorithms*, 41(1):99–113, 2001. 53

[Kol11] Alexandra Kolla. Spectral algorithms for unique games. *Comput. Complexity*, 20(2):177–206, 2011. 24, 27

[KS17a] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for minimum planarization (almost). In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pages 779–788. IEEE Computer Soc., Los Alamitos, CA, 2017. 63

[KS17b] Pravesh K. Kothari and David Steurer. Outlier-robust moment-estimation via sum-of-squares, 2017. 17

[KT07] Alexandra Kolla and Madhur Tulsiani. Playing random and expanding unique games, 2007. 24, 27

[Kuc95] Ludek Kucera. Expected complexity of graph partitioning problems. *Discret. Appl. Math.*, 57(2-3):193–212, 1995. 3, 47

[Las01] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2000/01. 17

[Lev18] Yevgeny Levanzov. On finding large cliques in random and semi-random graphs. Master's thesis, Weizmann Institute of Science, 1 2018. 22

[LG79] P. C. Liu and R. C. Geldmacher. On the deletion of nonplanar edges of a graph. In *Proceedings of the Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1979)*, Congress. Numer., XXIII–XXIV, pages 727–738. Utilitas Math., Winnipeg, Man., 1979. 64

[LOT12] James R. Lee, Shayan OveisGharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order Cheeger inequalities. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 1117–1130. ACM, New York, 2012. 38

[LRTV11] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. Algorithmic extensions of Cheeger's inequality to higher eigenvalues and partitions. In

Approximation, randomization, and combinatorial optimization, volume 6845 of *Lecture Notes in Comput. Sci.*, pages 315–326. Springer, Heidelberg, 2011. 43

[LRTV12] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. Many sparse cuts via higher eigenvalues. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 1131–1140. ACM, New York, 2012. 38

[LV18] Anand Louis and Rakesh Venkat. Semi-random graphs with planted sparse vertex cuts: algorithms for exact and approximate recovery. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 101, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018. 6, 17, 26

[LV19] Anand Louis and Rakesh Venkat. Planted models for k-way edge and vertex expansion. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 150 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 23, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. 6

[LY93] Carsten Lund and Mihalis Yannakakis. The approximation of maximum subgraph problems. In Andrzej Lingas, Rolf Karlsson, and Svante Carlsson, editors, *Automata, Languages and Programming*, pages 40–51, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. 63

[Man17] Pasin Manurangsi. Inapproximability of maximum edge biclique, maximum balanced biclique and minimum k-cut from the small set expansion hypothesis. In *44th International Colloquium on Automata, Languages, and Programming*, volume 80 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 79, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017. 21

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, pages 529–537. IEEE Computer Soc., Los Alamitos, CA, 2001. 23

[MM20] Konstantin Makarychev and Yury Makarychev. Certified algorithms: Worst-case analysis and beyond. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 49:1–49:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 2

[MMT20] Theo McKenzie, Hermish Mehta, and Luca Trevisan. A new algorithm for the robust semi-random independent set problem. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 738–746. SIAM, 2020. 5, 6, 8, 19, 53, 54, 55, 56, 59, 69

[MMV12] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 367–384. ACM, New York, 2012. 6, 54

[MMV14a] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Bilu-Linial stable instances of max cut and minimum multiway cut. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 890–906. ACM, New York, 2014. 2

[MMV14b] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the PIE model. In *STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing*, pages 41–49. ACM, New York, 2014. 6

[MNS12] Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction, 2012. 6

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing*. Cambridge University Press, Cambridge, second edition, 2017. Randomization and probabilistic techniques in algorithms and data analysis. 11

[Nes00] Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, volume 33 of *Appl. Optim.*, pages 405–440. Kluwer Acad. Publ., Dordrecht, 2000. 17

[NJW01] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 849–856. MIT Press, 2001. 38

[Par03] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003. 17

[Pee03] René Peeters. The maximum edge biclique problem is NP-complete. *Discrete Appl. Math.*, 131(3):651–654, 2003. 22

[Rot13] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms – Lecture Notes for the MAPSP Tutorial, 2013. https://sites.math.washington.edu/~rothvoss/lecturenotes/lasserresurvey.pdf. 9, 17, 55

[Rou21] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021. 2, 5, 69

[RS95] Neil Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *J. Combin. Theory Ser. B*, 63(1):65–110, 1995. 63

[Sho87] NZ Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. kibernetika 5 102–106.. 1998. *Nondifferentiable Optimization and Polynomial Problems*, 1987. 17

[SN97] Tom A. B. Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classification*, 14(1):75–100, 1997. 6

[ST01] Daniel Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 296–305. ACM, New York, 2001. 2

[Tre17] Luca Trevisan. Beyond worst-case analysis: Lecture 7, 10 2017. URL: https://lucatrevisan.wordpress.com/2017/10/13/beyond-worst-case-analysis-lecture-7/. 64, 65

[Tro12] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 12(4):389–434, 2012. 43

[TSS02] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:S136–44, 2002. 22

[Ver18] Roman Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018. An introduction with applications in data science, With a foreword by Sara van de Geer. 9, 11, 24, 30

[Vu07] Van H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007. 12

[Wig58] Eugene P. Wigner. On the distribution of the roots of certain symmetric matrices. *Ann. of Math. (2)*, 67:325–327, 1958. 12

[WS11] David P. Williamson and David B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, Cambridge, 2011. 9

[Yan78] Mihalis Yannakakis. Node-and edge-deletion np-complete problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, page 253–264, New York, NY, USA, 1978. Association for Computing Machinery. 8, 18, 63

[Zha08] Yun Zhang. Elissa j. chesler, michael a. langston: On finding bicliques in bipartite graphs: a novel algorithm with application to the integration of diverse biological data types. In *41st Hawaii International International Conference on Systems Science (HICSS-41 2008), Proceedings, 7-10 January 2008, Waikoloa, Big Island, HI, USA*, page 473. IEEE Computer Society, 2008. 18

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007. 52