

GRAPH PARTITIONING  
UNDER THE SPECTRAL LENS

KONSTANTINOS KOILIARIS

A DISSERTATION  
PRESENTED TO THE FACULTY  
OF MPLA  
IN CANDIDACY FOR THE DEGREE  
OF MASTER OF SCIENCE

RECOMMENDED FOR ACCEPTANCE  
BY THE DEPARTMENT OF  
MATHEMATICS  
OF THE  
UNIVERSITY OF ATHENS

ADVISER: PROFESSOR STATHIS ZACHOS  
EXT. PROFESSOR ALEXANDRA KOLLA

JANUARY 2015

© Copyright by Konstantinos Koiliaris, 2015.

All rights reserved.

# Abstract

Finding underlying structure in data has been a fundamental task for mathematicians, computer- and now data-scientists and the importance of clustering and partitioning data has increased dramatically in the past decade. We present the journey of one of the most essential problems in the area: Graph Partitioning. We begin with the importance and the wide range of applications it finds, the computational difficulties involved in solving it efficiently and the inapproximability results tied to it. We demonstrate the first average case analysis approaches using random models, the most prominent of which is the *planted partition model* or *stochastic block model* where the graph has  $k$  equally sized blocks and vertices connect independently with probability  $p$  within blocks and  $q$  across blocks. Recently, a large amount of research in computer science and statistics has been invested in providing lower-bounds on the scaling of  $|p - q|$  to ensure recovery of the planted blocks (partitions). We focus on the seminal results using spectral techniques and provide a high level overview of this rapidly evolving area, including the recent information-theoretic perspective threshold on the  $|p - q|$  range for recovery.

Finally, give our own spectral approach for solving Graph Partitioning for arbitrary  $k$  in the planted partition model and albeit not improving the state-of-the-art we believe our approach constitutes a new, easier proof for a very recent result.

## Acknowledgements

I would like to thank Prof. Stathis Zachos for being an exceptional teacher and an inspiring mentor throughout the last years of my undergraduate degree and again during my time at MPLA, but, more importantly, I want to thank him for creating a family environment and an ideal academic hub for me and all other theory students at NTUA. Moreover, I want to thank Prof. Dimitrios Thilikos for being the most charismatic professor I have ever had the honor of working with and for all the brainstorming sessions over Skype that we shared - I truly value them. I want to also thank Prof. Aris Pagourtzis for being there for me through the bad and the good years, and for his role in the aforementioned hub.

Without your support during my grad school applications, I would not have made it, thank you.

Finally, I want to thank Prof. Alexandra Kolla for believing in me. It was her guidance that lead me to work on this wonderful problem and introduced me to the spectral world, most of this work is a product of our collaboration up to now and none of this would have been possible without her.

*To my parents,  
for the love,  
they breathed in me.*

# Contents

Abstract . . . . .	iii
Acknowledgements . . . . .	iv
<b>1 Introduction</b>	<b>1</b>
1.0.1 Goal . . . . .	2
1.1 Preliminaries . . . . .	2
1.1.1 Objectives . . . . .	5
1.2 Tools . . . . .	5
1.3 Applications . . . . .	7
1.3.1 Parallel Processing . . . . .	7
1.3.2 VLSI Design . . . . .	8
1.3.3 Networks . . . . .	8
1.3.4 Image Processing . . . . .	9
<b>2 Graph Partitioning(<math>G, c(\cdot), w(\cdot), k</math>)</b>	<b>11</b>
2.1 Hardness . . . . .	12
2.2 Inapproximability . . . . .	14
2.3 A Different Approach . . . . .	16
2.3.1 The Planted Partition Model . . . . .	17
<b>3 A Spectral Lens</b>	<b>18</b>
3.1 Eigenvalues and Graph Bisection: An Average-case Analysis . . . . .	19

3.1.1	Description of the Algorithm . . . . .	20
3.1.2	Probabilistic Analysis . . . . .	22
3.2	Heuristics for Semirandom Graph Problems . . . . .	25
3.3	Spectral Partitioning of Random Graphs . . . . .	27
3.4	The Current State-of-the-Art . . . . .	28
3.4.1	Sharp Thresholds for Recovery in the Planted Partition Model	29
<b>4</b>	<b>A Different Spectral Approach</b>	<b>33</b>
4.0.2	The Planted Partition Model . . . . .	34
4.1	The SDP . . . . .	35
4.2	The Dual . . . . .	35
4.3	The Approach . . . . .	36
<b>5</b>	<b>Open Problems and Future Work</b>	<b>42</b>
	<b>Bibliography</b>	<b>44</b>

# Chapter 1

## Introduction

” *Quickly, bring me a beaker of wine, so that I may wet my mind and say something clever.*

”

---

Aristophanes, 450 - 385 B.C.

The procedure of categorizing data based on some metric has been an important part of science, since its inception. This problem of dividing or separating items to optimize a predetermined quantity is a classical one regardless of the way it is expressed, and it is one of the most basic algorithmic operations. Mathematicians throughout history encountered this problem in the form of separating numbers, sets and finally, since the 18th century, vertices in graphs - with the first results on Graph Theory.

Partitioning the vertices of a graph in order to minimize or maximize some property became a standard question in mathematics, and in the core of it was the most natural one: can we partition the vertices of a graph in such a way to minimize (maximize) the total number of edges going from one partition to another? Soon graphs became a standard abstraction tool for computer scientists and were widely embraced,



and the question of graph partitioning appeared both as a standalone problem and as a subproblem for more complicated applications.

### 1.0.1 Goal

This work aims at giving a thorough overview of the most popular graph partitioning setting, motivating and explaining the important applications it serves, highlighting the computational caveats involved in answering questions about it, and showcasing old and new spectral approaches for solving it.

## 1.1 Preliminaries

Let  $G = (V, E)$  be an *undirected graph* where  $|V| = n$  and  $|E| = m$  with *non-negative* edge weights  $w : E \rightarrow \mathbb{R}_{>0}$  and let  $k \in \mathbb{N}_{>1}$  be a number. We call  $\mathcal{P}$  a *partition* of  $G$  if it is a decomposition of its vertices  $V$  into disjoint nonempty blocks (subsets)  $\mathcal{P} = \{V_1, V_2, \dots, V_k\}$  such that:

$$V = \bigcup_{i \in [k]} V_i \tag{1.1}$$

and

$$V_i \cap V_j = \emptyset, \forall i \neq j. \tag{1.2}$$

The *cost*  $\mathcal{C}(\mathcal{P})$  of a partition  $\mathcal{P}$  is defined as the weighted sum of the edges connecting vertices belonging to distinct blocks. We will call the partition *balanced* when all the blocks have equal weights. In particular, we will quantify the notion of balance as follows: if  $(\forall i \in [k]) \quad [|V_i| \leq (1 + \varepsilon)\lceil n/k \rceil] =: L_{\max}$  for some *imbalance parameter*  $\varepsilon \in \mathbb{R}_{\geq 0}$ . If  $\varepsilon = 0$  we call the partition *perfectly balanced*. Sometimes we also use weighted vertices with vertex weights  $c : V \rightarrow \mathbb{R}_{>0}$ . Weight functions on vertices and edges are extended to sets of such objects by summing their weights. A block  $V_i$  is *overloaded* if  $|V_i| > L_{\max}$ . A *clustering* is also a partition of the vertices, however  $k$

will usually not be given and we will not require that it is balanced <sup>1</sup>. Note that a partition is also a clustering of a graph. In both cases, the *goal* is to minimize or maximize a particular objective function. An edge that runs between blocks is also called *cut edge*. The set  $E_{ij} = \{\{u, v\} \in E \mid u \in V_i \wedge v \in V_j\}$  is the set of cut edges between two blocks  $V_i$  and  $V_j$ . An abstract visualization of a partitioned graph is the *quotient meta-graph* where vertices represent blocks and edges are induced by connectivity between blocks. There is an edge in the quotient graph between blocks  $V_i$  and  $V_j$  if and only if there is an edge between a vertex in  $V_i$  and a vertex in  $V_j$  in the original, partitioned graph. A vertex  $v$  is a *neighbor* of vertex  $u$  if there is an edge  $\{u, v\} \in E$ . We will call  $v \in V_i$  a *boundary vertex* if it has a neighbor  $w \in V_j$  where  $i \neq j$ . The degree  $d(v)$  of a vertex  $v$  is the number of its neighbors. A cycle in a directed graph with negative weight is also called *negative cycle*. A *matching*  $M \subseteq E$  is a set of edges that do not share any common vertices, i.e., the graph  $(V, M)$  has maximum degree one.

An *adjacency matrix* of a graph  $G$  is a  $n \times n$  matrix  $A = A_G$ :

$$A_G = \begin{cases} w(u, v) & \text{if } (u, v) \in E \\ 0 & \text{if } (u, v) \notin E \end{cases} \quad (1.3)$$

If the graph has  $n$  vertices,  $A_G$  has  $n$  real *eigenvalues*  $\lambda_1(A_G) \geq \dots \geq \lambda_n(A_G)$  which we call the *spectrum* of  $A_G$ . The *eigenvectors* that correspond to these eigenvalues form an orthonormal basis of  $\mathbb{R}^n$ . Note that if the graph is *d-regular* then the largest eigenvalue is equal to  $d$  and the corresponding eigenvector is the all-one's vector, which we denote  $\mathbb{1}$ . We can use the *Courant-Fisher Theorem* to characterize the spectrum of  $A$ . The largest eigenvalue satisfies

$$\lambda_1(A_G) = \max_{x \in \mathbb{R}^n} \frac{x^T A x}{x^T x} \quad (1.4)$$

---

<sup>1</sup>More on clustering will be covered later.

If we denote the first eigenvector by  $x_1$  then

$$\lambda_2(A_G) = \max_{x \in \mathbb{R}^n, x \perp x_1} \frac{x^T A x}{x^T x} \quad (1.5)$$

Similar definitions hold for the eigenvalues  $\lambda_i$ ,  $i \geq 3$ . We will also need to define the *Laplacian matrix* of a graph  $G$  is defined as  $L = D - A$ , where  $D$  is the diagonal matrix with diagonal entry  $D(u, u) = d_u$  equal to the degree of vertex  $u$ , and  $A$  is the adjacency matrix. If the graph has  $n$  vertices,  $L_G$  has  $n$  real eigenvalues  $0 = \lambda_1(L_G) \leq \dots \leq \lambda_n(L_G)$  which we call the *spectrum* of  $L_G$ . We can always choose  $n$  eigenvectors  $\gamma_1, \dots, \gamma_n$  such that  $\gamma_i$  has eigenvalue  $\lambda_i$  which also form an orthonormal basis of  $\mathbb{R}^n$ . We note that 0 is always an eigenvalue with corresponding unit length eigenvector the (normalized) all-one's vector  $\mathbb{1}$ . Moreover, if and only if the graph has  $k$  connected components, then  $L_G$  has  $k$  eigenvalues equal to zero. We also define the *Normalized Laplacian matrix* to be the matrix:

$$\mathcal{L}_G = D^{-1/2} L_G D^{-1/2}. \quad (1.6)$$

An *Erdős-Rényi (ER) random graph*, denoted  $\mathcal{G}_{n,p}$ , is a graph composed of  $n$  vertices and for each pair of vertices, out of all the possible  $\binom{n}{2}$  pairs, there exists an edge with probability  $p$  independently. Another (ER) random graph variant, denoted  $\mathcal{G}_{n,m}$ , is a graph chosen uniformly at random from the collection of all graphs which have  $n$  vertices and  $m$  edges. Equivalently, all graphs of  $n$  vertices and  $m$  edges have equal probability of  $p^m (1-p)^{\binom{n}{2}-m}$ .

## Conventions

Let  $[n] = \{1, 2, \dots, n\}$ . All logarithms unless otherwise stated are base 2.

### 1.1.1 Objectives

As discussed in practice, one often seeks to find a partition that would minimize (or maximize) an objective. Depending on the different partition problems this can vary a lot. The most prominent objective function is to minimize the cost of the total cut of the partition  $\mathcal{P}$

$$\mathcal{C}(\mathcal{P}) = \sum_{i < j} w(E_{ij}). \quad (1.7)$$

Which is considered to be the standard formulation of the Graph Partitioning (GP) problem. But this is not the only popular variant, for instance when graph partitioning is used in parallel settings to map vertices to different sites (machines), the *communication volume* is more important than the cut itself [23]. Let  $D(v)$  denote the number of distinct blocks vertex  $v$  has neighbors in excluding his own block, then the communication volume is defined as  $\text{comm}(V_i) := \sum_{v \in V_i} c(v)D(v)$ . Naturally, the *maximum communication volume* is defined as  $\max \text{comm}(V_i)$  and the *total communication volume* as  $\sum_i \text{comm}(V_i)$ . The setting of maximum communication volume was used in a sub-challenge of the 10th DIMACS Challenge on Graph Partitioning and Graph Clustering [6].

Despite settings such as this one, traditionally minimizing the cut size has been the established standard for GP, both for practical reasons and because of how most of the time the cut correlates to other formulations. In this work we will only be concerned with graph partitioning where the function is minimizing the cut.

## 1.2 Tools

In this section we state a few results we will use later on. Depending on the familiarity of the reader some of these may be skipped.

We consider  $m \times n$  matrices over the real numbers. We are mostly looking at *square* matrices where  $m = n$ . Vectors  $x, y$  are orthogonal (denoted  $x \perp y$ ) if their inner product is zero, i.e.,  $x^T y = 0$ . If  $A$  is a square  $n \times n$  matrix with eigenvalues  $\lambda_1, \dots, \lambda_n$  then  $\text{tr}(A) = \sum_{i=1}^n \lambda_i$ .

An  $n \times n$  symmetric matrix  $A$  is said to be *positive semidefinite* if  $x^T A x \geq 0$  for all  $x \in \mathbb{R}^n$ .

**Theorem 1.2.1** (positive semidefinite matrices). *Let  $A$  be a real symmetric matrix. The following are equivalent:*

- *The matrix  $A$  is positive semidefinite.*
- *All eigenvalues of  $A$  are nonnegative.*

We will be using a few variations of *Chernoff Bounds*.

**Theorem 1.2.2** (Chernoff Bound 1). *Let  $X \sim \text{Binomial}(n, 1/2)$ . Then for any  $0 \leq t \leq \sqrt{n}$ ,*

$$\Pr \left[ X \geq \frac{n}{2} + t \frac{\sqrt{n}}{2} \right] \leq e^{-t^2/2}, \quad (1.8)$$

$$\text{and also } \Pr \left[ X \leq \frac{n}{2} - t \frac{\sqrt{n}}{2} \right] \leq e^{-t^2/2} \quad (1.9)$$

**Theorem 1.2.3** (Chernoff Bound 2). *Let  $X_1, \dots, X_n$  be independent random variables (they need not necessarily have the same distribution). Assume that  $0 \leq X_i \leq 1$ , for every  $i \in [n]$ . Let  $X = X_1 + \dots + X_n$ . We write  $\mu = \mathbf{E}[X] = \mathbf{E}[X_1] + \dots + \mathbf{E}[X_n]$ . Then for any  $\varepsilon \geq 0$*

$$\Pr [X \geq (1 + \varepsilon)\mu] \leq \exp \left( -\frac{\varepsilon^2}{2 + \varepsilon} \mu \right), \quad (1.10)$$

$$\text{and } \Pr [X \leq (1 - \varepsilon)\mu] \leq \exp \left( -\frac{\varepsilon^2}{2} \mu \right) \quad (1.11)$$

Another result that will be used is the *Matrix Bernstein* inequality.

**Theorem 1.2.4** (Matrix Bernstein). *Consider a finite sequence  $X_i$  of independent, random, Hermitian matrices with dimension  $d$ . Assume that*

$$\mathbf{E}[X_i] = 0 \text{ and } \lambda_{\max}(X_i) \leq R. \quad (1.12)$$

*Introduce the random Matrix*

$$X = \sum_{i=1}^n X_i. \quad (1.13)$$

*Compute the variance parameter*

$$\sigma^2 = \sigma^2(X) = \|\mathbf{E}[X^2]\|. \quad (1.14)$$

*Then*

$$\mathbf{E}[\lambda_{\max}(X)] \leq \sqrt{2\sigma^2 \log d} + \frac{1}{3}R \log d. \quad (1.15)$$

*Furthermore, for any  $t \geq 0$ :*

$$\Pr[\lambda_{\max}(X) \geq t] \leq d \exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right). \quad (1.16)$$

## 1.3 Applications

The question of partitioning graphs was one that appeared often in practice and as the problems we modeled increased in size so did their abstractions, with applications in software and hardware design, networks, road design, image processing, biology and more. In this section we go through some of the applications of graph partitioning.

### 1.3.1 Parallel Processing

Perhaps the canonical application of graph partitioning is the distribution of work to processors of a parallel machine. Scientific computing applications such as sparse

direct and iterative solvers extensively use graph partitioning to ensure load balance and minimize communication. When the problem domain does not change as the computation proceeds, graph partitioning can be applied once in the beginning of the computation. This is known as static partitioning.

### **1.3.2 VLSI Design**

Physical design of digital circuits for very large-scale integration (VLSI) systems has a long history of being one of the most important customers of graph and hypergraph partitioning, often reinforced by several additional domain relevant constraints. The partitioning should be accomplished in a reasonable computation time, even for circuits with millions of modules, since it is one of the bottlenecks of the design process. The goal of the partitioning is to reduce the VLSI design complexity by partitioning it into smaller components (that can range from a small set of field-programmable gate arrays to fully functional integrated circuits) as well as to keep the total length of all the wires short. The typical optimization objective is to minimize the total weight of connections between subcircuits (blocks), where vertices are the cells, i.e., small logical or functional units of the circuit (such as gates), and edges are the wires. Because the gates are connected with wires with more than two endpoints, hypergraphs model the circuit more accurately. Examples of additional constraints for the VLSI partitioning include information on the I/O of the circuit, sets of cells that must belong to the same blocks, and maximum cut size between two blocks. For more information about partitioning of VLSI circuits see [27].

### **1.3.3 Networks**

In addition to the previously mentioned task of network data distribution across a cluster of machines for fast parallel computations, complex networks introduced numerous further applications of graph partitioning. A common task in these ap-

plications is to identify groups of similar entities whose similarity and connectivity is modeled by the respective networks. The quality of the localizations is quantified with different domain-relevant objectives. Many of them are based on the principle of finding groups of entities that are weakly connected to the rest of the network. In many cases such connectivity also represents similarity. In the context of optimization problems on graphs, by complex networks we mean weighted graphs with non-trivial structural properties that were created by real-life or modeling processes. Often, models and real-life network generation processes are not well understood, so designing optimization algorithms for such graphs exhibit a major bottleneck in many applications.

### **1.3.4 Image Processing**

Image segmentation is a fundamental task in computer vision for which graph partitioning and clustering methods have become among the most attractive solution techniques. The goal of image segmentation is to partition the pixels of an image into groups that correspond to objects. Since the computations preceding segmentation are often relatively cheap and since the computations after segmentation work on a drastically compressed representation of the image (objects rather than pixels), segmentation is often the computationally most demanding part in an image processing pipeline. The image segmentation problem is not well-posed and can usually imply more than one solution. During the last two decades, graph-based representations of an image became very popular and gave rise to many cut-based approaches for several problems including image segmentation. In this representation each image pixel (or in some cases groups of pixels) corresponds to a vertex in a graph. Two vertices are connected by a weighted edge if some similarity exists between them. Usually, the criteria of similarity is a small geodesic distance which can result in mesh-like graphs with four or more neighbors for each vertex. The edge weights represent another mea-



sure of (dis)similarity between vertices such as the difference in the intensity between the connected pixels (vertices).

Naturally this is not a complete exhaustive list of applications, but a description of the most typical ones. The problem has dozens of applications and it would be impossible to include them all here. As mentioned before, the importance of Graph Partitioning does not lie only on the problems that it directly models (applications) but also and possibly more substantially in the fact that it is a primitive algorithmic procedure (tool) used as a subroutine in numerous more complicated problems and techniques, such as *divide-and-conquer algorithms*.

# Chapter 2

## Graph Partitioning( $G, c(\cdot), w(\cdot), k$ )

” *In mathematics the art of proposing a question must be held of higher value than solving it.* ”

---

Georg Cantor, 1845 - 1918

At this point we state formally the most general setting of the GRAPH PARTITIONING problem:

GRAPH PARTITIONING( $G, c(\cdot), w(\cdot), k$ )

Given an undirected graph  $G = (V, E)$ , non-negative weight functions on the vertices  $c : V \rightarrow \mathbb{R}_{>0}$  and edges  $w : E \rightarrow \mathbb{R}_{>0}$  and an integer  $k \geq 2$ , find a perfectly balanced partition  $\mathcal{P}$  of  $V$  into  $k$  blocks  $\{V_i\}_{i=1}^k$  of at most  $\lceil \sum_{v \in V_i} c(v)/k \rceil$  vertex weight that minimizes the total weight of the edges that connect distinct blocks.

For  $k = 2$  the problem reduces to finding a partition into two blocks of equal weight that minimizes the weight of the edges across them, which is also known as the MIN BISECTION problem. In the case of arbitrary  $k$  the problem is also known as

MIN MULTISECTION. The problems remain hard in the unweighted case (or simply  $\forall e \in E : w(e) = 1$ ) as well.

This is a good point to make a small parenthesis and notice the differences between the GRAPH PARTITIONING and other vertex partitioning problems<sup>1</sup>. The  $k$ -COLORING problem asks to partition the vertices into  $k$  blocks (colors) not necessarily balanced, such that no edge within a block exists - instead of minimizing cut edges. The CLUSTERING problem on the other hand does not even fix the number of blocks before hand, nor does it require that they are balanced, you just try to find any partition that minimizes the edges cut. A dual problem to GRAPH PARTITIONING in a way is the MAX CUT problem, where we want to partition the vertices in to two sets in order to maximize the number of edges in the cut.

This general version of the problem: GRAPH PARTITIONING( $G, c(\cdot), w(\cdot), k$ ) can be simplified by assuming that  $\forall v \in V : c(v) = 1$  and  $\forall e \in E : w(e) = 1$  without reducing its difficulty and from this point on, we will be using the more general weighted setting only when necessary.

## 2.1 Hardness

In October of '73 Hyafil and Rivest showed in [24] that (the decision version of) GRAPH PARTITIONING is hard:

**Theorem 2.1.1** (Hyafil and Rivest '73 [24]). *The problem of answering if an undirected, edge-weighted graph  $G$  can be partitioned into blocks of at most  $r$  vertices with a total weight of cut edges less than or equal to  $W$  is **NP**-complete.*

*Proof.* The **NP** membership of GP should be clear: given a succinct candidate solution, which is simply a mapping of the vertices into the different blocks, one has to

---

<sup>1</sup>The sister problem known as PARTITIONING - where we partition numbers - interestingly enough is called "The Easiest Hard Problem".

verify that each block is bounded in size by  $r$  and that the total weight of the edges in any block does not surpass  $W$ .

To show **NP** completeness we will reduce from EXACTLY-3-SAT. Construct a graph  $G$  from the 3CNF formula  $\Phi$  with set of clauses  $C = \{c_i\}_{i=1}^r$  in two steps as follows.

First we consider the graph  $G'$  that has  $3r$  vertices, one for every literal of every clause and two vertices are connected with an edge when the literals they represent are in distinct clauses and aren't complementary. Notice that  $G'$  has a clique of size  $r$  if and only if  $\Phi$  is satisfiable.

We now modify  $G'$  by adding to every triplet of vertices corresponding to a single clause, a clique of size  $(r - 2)$  and connect it completely with the three existing vertices, call the resulting graph  $G$ . Observe that this construction of  $G$  can easily be completed in polynomial time.

Now,  $G$  contains a clique of size  $r$  if and only if  $G'$  does, that is if and only if  $\Phi$  is satisfiable. Set  $W := |E| - (r + 1) \cdot \binom{r}{2} + r$  then the claim is that  $\text{GP}(G, r, W)$  has a solution if and only if  $\Phi$  is satisfiable.

**Claim 2.1.2.** *In every optimal partition the clique of size  $r$  will be contained as a distinct block, if one exists.*

*Proof.* We discern two cases based on the existence of this clique.

- Assuming that the clique of size  $r$  exists, let  $\mathcal{P}^*$  be the partition which has it as a distinct block and also has  $r$  other blocks, each containing one smaller clique of size  $(r - 2)$  attached to two of the three vertices connected to that clique. Recall that the clique of size  $r$  contains exactly one point from every triplet of vertices corresponding to a clause. Then the total weight of this partition is

exactly:

$$|E| - \binom{r}{2} - r \left[ \binom{r}{2} - 1 \right] = |E| - (r+1) \binom{r}{2} + r = W. \quad (2.1)$$

- If no clique of size  $r$  exists in  $G$ , then the average number of edges per vertex within a block can be at most  $\left[ \binom{r}{2} - 1 \right] / r$ . This is true because this is the average number of edges per vertex in a clique of size  $r$  when it is lacking one edge. In which case, the total weight of the partition is greater than:

$$|E| - (r+1) \left[ \binom{r}{2} - 1 \right] > W. \quad (2.2)$$

Thus the optimal partition contains a clique of size  $r$  of  $G$  as a separate block, if it exists. Any other partition  $\mathcal{P}'$  such that  $\mathcal{C}((P')) \leq \mathcal{C}((P^*))$  would also have to contain a clique of size  $r$  as a block, since the number of internal edges will be sufficiently high.  $\square$

Thus we have shown that  $\text{GP} \in \mathbf{NP}$  and  $\text{GP} \leq_{\mathbf{P}} \text{EXACTLY-3-SAT}$ , hence  $\text{GP}$  is  $\mathbf{NP}$ -complete.  $\square$

An immediate corollary of the above hardness result is the hardness characterization for the more general weighted case.

**Corollary 2.1.3.** *The general (decision) version of the GRAPH PARTITIONING( $G, c(\cdot), w(\cdot), k, W$ ) is  $\mathbf{NP}$ -complete.*

## 2.2 Inapproximability

Encountering problems that are very useful in practice but present computational difficulties is not a rare occasion in Computer Science, the class of  $\mathbf{NP}$ -complete problems is filled with such cases. So, do we just give up? No, never! The fact

that we can not, unless Hell freezes over<sup>2</sup>, solve these problems *exact* in polynomial time, does not mean that the problem is unsolvable in polynomial time. In many cases of hard problems, we have managed to find polynomial time algorithms that *approximate* the solution to almost arbitrary precision.

Well, sadly this will not be the case for GRAPH PARTITIONING as Andreev and Räcke in 2004 shot down any hopes for an efficient polynomial time approximation algorithm for the popular partitioning problem in [3].

**Theorem 2.2.1** (Andreev and Räcke '04 [3]). *The  $(k, 1)$ -balanced partitioning problem -  $GP(G, w(\cdot), k, 0)$  has no polynomial time approximation algorithm with finite approximation factor unless  $\mathbf{P} = \mathbf{NP}$ .*

*Proof.* We use a reduction from the 3-PARTITION problem defined as follows. Given  $n = 3k$  integers  $\{a_i\}_{i=1}^n$  and a threshold number  $A$  such that  $\frac{A}{4} < a_i < \frac{A}{2}$  and

$$\sum_{i=1}^n a_i = k \cdot A. \tag{2.3}$$

The task is to decide if the numbers can be partitioned into triples such that each triple adds up to  $A$ . This problem is *strongly*  $\mathbf{NP}$ -complete [18]. Hence it is  $\mathbf{NP}$ -complete even in the case that all numbers  $(\{a_i\}_{i=1}^n, A)$  are polynomially bounded.

Suppose that we have a polynomial time approximation algorithm for  $(k, 1)$ -balanced partition with finite approximation factor. We can use this algorithm to solve an instance of 3-PARTITION with polynomially bounded numbers, in the following manner.

We construct a graph  $G$  such that for each number  $a_i$  it contains a clique size  $a_i$ . This construction can be concluded in polynomial time only because 3-PARTITION is strongly  $\mathbf{NP}$ -complete and so all its numbers are polynomially bounded in the

---

<sup>2</sup>Yes, it is a strong opinion of the author that  $\mathbf{P} \neq \mathbf{NP}$ !

length of the input (Otherwise the above graph would not be polynomially time constructible).

If the 3-PARTITION instance can be solved, the  $(k, 1)$ -balanced partition problem in  $G$  can be solved without cutting any edge. On the contrary, if the 3-PARTITION instance cannot be solved, the optimum  $(k, 1)$ -balanced partitioning in  $G$  will cut at least one edge. But an approximation algorithm with finite approximation factor has to differentiate between these two cases - which is not the case here. Hence, it can solve the 3-PARTITION problem which is a contradiction under the assumption that  $\mathbf{P} \neq \mathbf{NP}$ . □

Currently the best known (real) approximation bound is due to Feige and Krauthgamer from their seminal paper [16], where they showed a bound of  $O(\log^2 n)$ . For special graph classes there have been a number of better results, including some described in [16], and even a PTAS by Arora et al. [4] for graphs of minimum degree  $\Omega(n)$ . Moreover, there are a number of polynomial (and not) bicriteria approximation algorithms for different values of the imbalance parameter  $\varepsilon$ , for more information on these results we refer the reader to [32], [14] and [3].

## 2.3 A Different Approach

So, where do we stand? Do we give up on all hope for an “accurate” solution to the balanced GRAPH PARTITIONING problem? Sure, the general version of the problem is hard, apparently too hard even, but does that mean that we have no hope for anything better?

Luckily for me, no. There is (as always) another approach; namely to look at the *Average Case analysis* for GRAPH PARTITIONING. Sure, the problem in the worst case looks very hard, but in practice how often do we encounter these bad instances?

How probable is it that the data scientists encounter these “worst case” instances with real-world data? Is the worst case performance indeed the norm or the exception?

Average Case analysis is equivalent to considering the probability of success of an algorithm on a random graph. However choosing graphs too uniformly from the set of all possible graphs, is not very helpful. Only a small percentage of all  $n$  vertex graphs have meaningful partitions, for this reason this new direction restricts the search space from all possible graphs on  $n$  vertices to graphs that, in expectation, will have meaningful partitions.

### 2.3.1 The Planted Partition Model

In '87 Boppana made the first step towards such an approach by introducing a *random graph model* to work on the problem of BISECTION his work in [7]. His model was defined as follows.

He assumed some initial bisection of the vertices into  $V_1$  and  $V_2$  and then for any two vertices inside a partition he sampled an edge independently at random with probability  $p$  and for any two vertices in distinct partitions he sampled an edge independently at random with probability  $q$  smaller than that of  $p$ . This way depending on the range of the  $p - q$  parameters, at least in expectation, there would be at least one meaningful cut in this graph, the *planted* one. Later on, this model would be named the *planted partition model* by computer scientists and *stochastic block model* by mathematicians.

Since '87, a variety of different random and semirandom models have been introduced [28], [15], [30], [20] for GRAPH PARTITIONING and other partitioning problems but the *planted partition model* has remained the most prominent one. In the following section we present the key results using this model for both the case of two and  $k$  blocks.



# Chapter 3

## A Spectral Lens

” *One person’s craziness is another person’s reality.*

”

---

Tim Burton, *1958* -

The word “spectrum” means different things in different areas of mathematics. For those of us that work in graph theory, the word denotes the eigenvalues of one of the several matrices associated with a graph.

Spectral graph theory studies precisely these connections between combinatorial properties of graphs and the eigenvalues of matrices associated to the graph, such as the adjacency matrix and the Laplacian matrix. These eigenvalues carry much information about a graph’s structure and properties.

Spectral graph theory has applications to the design and analysis of approximation algorithms for graph partitioning problems, to the study of random walks in graph, to the construction of expander graphs, and more. It also reveals connections between the above topics, and provides, for example, a way to use random walks to approximately solve graph partitioning problems.

One of the first contributions of spectral graph theory in algorithms happened in '94 when Goemans and Williamson in [19] gave a breakthrough 0.878-approximation algorithm for MAX CUT, improving significantly upon the previously (agnostic) approximation algorithm that had a performance guarantee of only 0.25. Their approach can be interpreted both as a semidefinite programming one and as an eigenvalue minimization one.

In the past two decades spectral techniques have become a standard tool in combinatorial optimization results [19], [15], [30], the study of explicit constructions of expander graphs [2], the unique games conjecture [5] and more [9]. In this chapter we will go through the biggest advances using spectral techniques for GRAPH PARTITIONING.

In the rest of this chapter, we will look at three pivotal spectral results [7], [15], [30] for graph partitioning and then discuss the current state-of-the-art and the latest advances.

### **3.1 Eigenvalues and Graph Bisection: An Average-case Analysis**

We begin with the first result by R.B. Boppana in '87 [7]. In this work it is the first time that one of the graph partition problems, GRAPH BISECTION, is analyzed under an average-case analysis and the result is a very influential one, signaling the start of a series of works on similar techniques for many more graph partitioning problems.

In his paper he presents an algorithm that will, for almost all graphs in a certain class (model), output the minimum-size bisection. Furthermore the algorithm will yield, for almost all such graphs, a proof that the bisection is optimal. The algorithm is based on computing eigenvalues and eigenvectors of matrices associated with the graph.

An average-case analysis must specify an appropriate probability distribution on the set of inputs. There is a standard probability distribution on the set of graphs, denoted  $\mathcal{G}_{n,m}$ , that is defined as follows. The vertex set  $V = \{1, 2, \dots, n\}$ . The probability distribution  $\mathcal{G}_{n,m}$  is uniform on the set of all graphs on  $V$  with  $m$  edges. Unfortunately, all the bisections of a random graph from  $\mathcal{G}_{n,m}$  will, with high probability, have size asymptotic to  $m/2$  as the ration  $m/n$  increases. To overcome this problem, we modify the probability distribution in one feature. The vertex set still remains  $V = \{1, 2, \dots, n\}$  for some integer  $n$ . The probability distribution used by Boppana, denoted  $\mathcal{G}_{n,m,b}$ , is uniform on the set of graphs on  $V$  vertices,  $m$  edges and bisection width of  $b$ . The bisection width  $b$  is chosen to be smaller than  $m/2$ . This additional condition ensures that at least one bisection is sufficiently smaller than an average bisection of the graph.

### 3.1.1 Description of the Algorithm

First some small definitions. As per usual we assume an undirected graph  $G = (V, E)$  with an even number of vertices  $n$  and without loss of generality assume that  $V = \{1, 2, \dots, n\}$ . Given a bisection  $(V_1, V_2)$  of  $G$  its *associated vector* has  $i$ th coordinate equal to 1 if  $i \in V_1$ , and equal to -1 if  $i \in V_2$ . Given two vectors  $d$  and  $x$  both in  $\mathbb{R}^n$ , define the function  $f$  equal to

$$f(G, d, x) := \sum_{\{i,j\} \in E} \frac{1 - x_i x_j}{2} + \sum_{i \in V} d_i (x_i^2 - 1). \quad (3.1)$$

We connect this function  $f$  with the bisections as follows:

**Lemma 3.1.1** ([7]). *Suppose  $(V_1, V_2)$  is bisection of a graph  $G = (V, E)$ , with associated vector  $x$ . Then for every vector  $d$ , the size of the bisection  $(V_1, V_2)$  equals  $f(G, d, x)$ .*

*Proof.* Notice that  $(1 - x_i x_j)/2$  is 1 if  $\{i, j\} \in E$  and 0 otherwise. Thus, the first sum in the expression for  $f$  above equals the size of the bisection  $(V_1, V_2)$  and the second sum is zero, since the coordinates of the vector  $x$  are  $\pm 1$ .  $\square$

We call  $S$  the subspace  $\{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 0\}$ . Let the *norm* of  $x$  be equal to  $\|x\| = (\sum_{i=1}^n x_i^2)^{1/2}$ . Define  $g$  equal to

$$g(G, d) := \min_{\substack{x \in S \\ \|x\| = \sqrt{n}}} f(G, d, x). \quad (3.2)$$

We will compute the value of  $g(G, d)$  using eigenvalues as described below.

As per usual  $A$  will be the adjacency matrix of  $G$ , given a vector  $d$ , let  $D = \text{diag}(d)$  be the diagonal matrix whose diagonal entries are the corresponding components of  $d$ . Let  $B = A + D$ . Put  $\text{sum}(B)$  equal to the sum of all  $n^2$  entries of  $B$ . Let  $B_S$  be the linear operator from  $S$  to  $S$  that maps  $x$  to the projection of  $Bx$  onto  $S$  (the closest point on  $S$  from  $Bx$ ). Finally, put  $\lambda(B_S)$  equal to the largest eigenvalue of  $B_S$ . Then  $g$  can be expressed as

$$g(G, d) = \frac{\text{sum}(B) - n\lambda(B_S)}{4}, \quad (3.3)$$

so  $g$  can be computed to arbitrary precision in polynomial time using standard eigenvalue procedures.

Given a graph  $G$ , set  $h$  equal to

$$h(G) := \max_{d \in \mathbb{R}^n} g(G, d). \quad (3.4)$$

Notice that  $g(G, d)$  is a concave function of  $d$ , so  $h(G)$  is the maximum of a concave function and under very general conditions the maximum of a concave function can be computed to arbitrary precision in polynomial time using the *ellipsoid method* [21].

Thus  $h(G)$  is computable to arbitrary precision in polynomial time. The theorem below shows that  $h(G)$  provides a lower bound on the bisection width of a graph  $G$ .

**Theorem 3.1.2** ([7]). *Every graph  $G$  has a bisection width of at least  $h(G)$ .*

*Proof.* Choose an arbitrary vector  $d$ . Suppose that  $(V_1, V_2)$  is the minimum-size bisection of  $G$ , with associated vector  $x$ . By Lemma 3.1.1 it follows that the bisection width of  $G$  equals  $f(G, d, x)$ . Notice that the vector  $x$  is in  $S$  and has norm  $\sqrt{n}$ , so by definition of  $g$ , the bisection width of  $G$  is at least  $g(G, d)$ . Since the vector  $d$  was chosen arbitrarily, the bisection width of  $G$  is at least  $h(G)$ .  $\square$

So far, only the lower bound  $h(G)$  has been described. The upper bound algorithm is as follows:

Given a graph  $G$  with adjacency matrix  $A$ , find the vector  $d$  that maximizes  $g(G, d)$ , using the methods of [21]. Let  $y$  be the eigenvector corresponding to the largest eigenvalue of  $(A + D)_S$ . Output the bisection that has the  $n/2$  largest components of  $y$  on one side, and the  $n/2$  smallest components on the other.

The next section is devoted on the correctness of the algorithm.

### 3.1.2 Probabilistic Analysis

Here, we provide the average-case analysis performance of the algorithm described in the previous section. Recall the model of random graphs  $\mathcal{G}_{n,m,b}$  defined earlier. The following theorem, the main result of this paper, shows that  $h(G)$  equals, with high probability, the bisection width of  $G$ . If the bisection width of  $G$  equals  $h(G)$ , then the upper bound algorithm of the previous section will actually find the minimum-size bisection. Thus, the following analysis will focus only on the lower bound  $h(G)$ .

**Theorem 3.1.3** ([7]). *Suppose that  $G$  is a random graph from  $\mathcal{G}_{n,m,b}$ , and that*

$$b \leq \frac{1}{2}m - \frac{5}{2}\sqrt{mn \log n}. \quad (3.5)$$

Then with probability  $1 - O(1/n)$ , the bisection width of  $G$  equals  $h(G)$ .

*Proof.* Call  $(V_1, V_2)$  the minimum-size bisection of  $G$ . For  $i$  in  $V_1$ , let  $d_i$  be defined as: ( $\#$  of vertices in  $V_2$  adjacent to  $i$ ) - ( $\#$  of vertices in  $V_1$  adjacent to  $i$ ). For  $i$  in  $V_2$  let  $d_i$  be the negation of this expression. By the definition of  $h$ , the equality  $g(G, d) = b$  implies that  $h(G) = b$ . Thus, it suffices to show that  $g(G, d) = b$  with high probability.

In the previous section, the function  $g$  was characterized in terms of eigenvalues:  $g(G, d) = b$  which is equivalent to  $\lambda(B_S) = 0$ . Let  $y$  be the associated vector  $(V_1, V_2)$ . Notice that  $B_S$  has an eigenvalue 0 with corresponding eigenvector  $y$ . It thus suffices to show that all other eigenvalues of  $B_S$  are nonpositive with high probability.

Consider the expected value  $\mathbf{E}[B]$  of the random matrix  $B = A + D$ . Let

$$p = \frac{(m-b)}{\frac{n}{2} \left(\frac{n}{2} - 1\right)} \quad \text{and} \quad q = \frac{b}{\left(\frac{n}{2}\right)^2}. \quad (3.6)$$

Let  $M$  be the matrix such that

$$M(i, j) = \begin{cases} p & \text{if } \{i, j\} \in V_k, k = 1, 2 \\ q & \text{otherwise} \end{cases} \quad (3.7)$$

Then  $\mathbf{E}[B] = M - \frac{1}{2}(p-q)nI$ , where  $I$  is the identity matrix. Notice that the operators  $B_S$  and  $(B - M)_S$  are equivalent on vectors that are orthogonal to  $y$ . Thus it suffices to show that all the eigenvectors of  $(B - M)_S$  are nonpositive with high probability. In turn, it suffices to show that all the eigenvalues of  $B - \mathbf{E}[B]$  are bounded above by  $\frac{1}{2}(p - q)n$  with high probability.

So far, the proof has dealt with a random graph  $G$  from the model  $\mathcal{G}_{n,m,b}$ . It is now convenient to replace this model with a closely related model. The new model will form a random graph as follows. First the vertices of the graph are randomly partitioned into two equal-size pieces. Each edge within either of the two blocks is

sampled independently with probability  $p$ , and each edge between the two blocks is sampled independently with a smaller probability  $q$ . Such a random graph is likely to have about  $m$  edges and bisection width about  $b$ . Thus it looks similar to a graph from  $\mathcal{G}_{n,mb}$ .

To estimate the eigenvalues of the matrix  $B - \mathbf{E}[B]$ , decompose the matrix into the two matrices  $A - \mathbf{E}[A]$  and  $D - \mathbf{E}[D]$ . We use the inequality

$$\lambda(B - \mathbf{E}[B]) \leq \lambda(A - \mathbf{E}[A]) + \lambda(D - \mathbf{E}[D]). \quad (3.8)$$

The eigenvalues of the matrix  $D - \mathbf{E}[D]$  are precisely its diagonal entries, and they can be shown to be small with high probability using Chernoff's bound on the tail of a binomial distribution. We obtain  $\lambda(D - \mathbf{E}[D]) \leq 5\sqrt{pn \log n}$  with high probability. To estimate the eigenvalues of  $A - \mathbf{E}[A]$ , the following theorem is used.

**Theorem 3.1.4** ([17]). *Let  $Z$  be a random  $n \times n$  symmetric matrix whose entries are independent up to symmetry, have mean zero, have standard deviation at most  $\sigma$ , and are concentrated on  $[-a, a]$ . Suppose that  $\sigma\sqrt{n} \geq 10a\sqrt{\log n}$ . Then with probability at least  $1 - O(1/n)$ , all the eigenvalues of  $Z$  have absolute value less than  $3\sigma\sqrt{n}$ .*

Applying this theorem to  $Z = A - \mathbf{E}[A]$ , it follows that  $\lambda(A - \mathbf{E}[A]) \leq 3\sqrt{pn}$  with high probability. Together with the above bound on  $\lambda(D - \mathbf{E}[D])$ , we obtain  $\lambda(B - \mathbf{E}[B]) \leq 6\sqrt{pn \log n}$ . But the hypothesis on the size of  $b$  implies that  $6\sqrt{pn \log n} < \frac{1}{2}(p - q)n$ . Thus the eigenvalues of  $B - \mathbf{E}[B]$  are less than  $\frac{1}{2}(p - q)n$  with high probability, which by the above remarks suffices to establish the theorem.  $\square$

In later revisions of this work, more emphasis is given on the model, as it attracted more and more attention.

## 3.2 Heuristics for Semirandom Graph Problems

In this work [15] Feige and Kilian consider a semirandom graph model for computing the problem of BISECTION. Unlike Boppana’s approach, Feige and Kilian blend random models with adversarial decisions. In particular, they use the random model introduced by [7]: they begin by choosing a random bisection  $(S, \bar{S})$  of the vertices and then each edge  $\{u, v\} \in S \times \bar{S}$  is independently sampled with probability  $q$  and each edge  $\{u, v\} \notin S \times \bar{S}$  is independently chosen with probability  $p > q$  forming the graph  $G_{rand}$ . On top of that, the adversary may then arbitrarily remove edges in  $S \times \bar{S}$  and add edges not in  $S \times \bar{S}$ , forming the final graph  $G$ .

As before we assume that  $G = (V, E)$  is a graph with even vertices  $n$ . The bisection width of  $G$  will be denoted as  $b(G)$  and will be equal to

$$b(G) := \min_S [|E(S, \bar{S})|]. \quad (3.9)$$

Let  $h(G)$  be an arbitrary function on graphs. The aim is to have  $h$  provide a heuristic for graph bisection in the sense that for many graphs  $h(G) = b(G)$ . Given the following semi-definite relaxation of bisection: for a graph  $G = (V, E)$  find an order  $n$  matrix  $X = \{x_{ij}\}$  such that

1.  $\forall i, x_{ii} = 1$ ,
2.  $\sum_{ij} x_{ij} = 0$ ,
3. The matrix  $X$  is symmetric and positive semidefinite,

Let  $h$  be given by the solution of this SDP

$$h(G) := \min_X h_X(G) = \min_X \sum_{\substack{x \in S \\ \|x\| = \sqrt{n}}} \frac{1 - x_{ij}}{2} \quad (3.10)$$



At this point Feige and Kilian prove that  $h$  is *robust* in the sense that a monotone adversary is powerless against the function: whenever  $G_{rand}$  is such that  $h$  computes the exact value of  $b(G_{rand})$ , the monotone adversary cannot prevent  $h$  from giving the value of  $b(G)$ . This way the analysis can completely ignore the existence of the adversary from this point on, significantly simplifying things.

So, to compute the bisection, we want the robust function  $h$  to be polynomially computable and with high probability:

$$h(G_{rand}) = b(G_{rand}). \tag{3.11}$$

This last property 3.11 of  $h$  we will call “probably good”. By choosing  $h$  to be the solution of an SDP we can then use the ellipsoid method to compute it within arbitrary precision limits in polynomial time, we know it is robust, so it remains to be shown that it is good estimate of bisection w.h.p., i.e., that it is “probably good.”

Taking advantage of the standard duality theory that exists in SDPs, similarly to linear programming, they take the dual:

$$\max \frac{m}{2} + \frac{1}{4} \sum_{i=1}^n y_i \tag{3.12}$$

subject to the constraint that the matrix  $M = -A - y_0 J - Y$  is positive semidefinite. Where  $m$  is the total number of edges in the graph,  $A$  is its adjacency matrix,  $J$  is the all 1 matrix of order  $n$ ,  $y_0$  is an auxiliary variable that only affects feasibility, but does not appear in the objective function, and finally  $Y$  is a diagonal matrix with the  $y_i$ 's along its diagonal.

**Theorem 3.2.1** ([15]). *For a sufficiently large constant  $c$ , the function  $h$  defined above is “probably good” when*

$$p - q \geq c \sqrt{p \frac{\log n}{n}}. \tag{3.13}$$

That is, with high probability,  $h(G_{rand}) = b(G_{rand})$ , where the probability is taken over the choice of  $G_{rand}$ .

*Proof.* We will show that if  $p - q$  is sufficiently large, then with high probability over the choice of  $G_{rand}$ , the dual semidefinite maximization problem has a feasible solution with value  $b(G_{rand})$ .

We now guess a feasible solution for the semidefinite maximization problems. For every  $i \in [n]$ , let  $y_i$  be the *contribution* of vertex  $i$  to the bisection  $(S, \bar{S})$ , namely,  $y_i$  is the difference between the number of edges vertex  $i$  has in the cut and the number of edges it has inside its own block.

The value of the objective function of the maximization semidefinite program is then

$$\frac{m}{2} + \frac{\sum_{i=1}^n y_i}{4} = \frac{m}{2} + \frac{2|(S, \bar{S})| - 2(m - |(S, \bar{S})|)}{4} = |(S, \bar{S})| \quad (3.14)$$

which is  $b(G_{rand})$ , as desired.

It remains to be seen that w.h.p., it is possible to choose  $y_0$  such that the matrix  $M = -A - y_0 J - Y$  is positive semidefinite. We shall choose  $y_0 = -1$  and show that the matrix  $M$  has no negative eigenvalues, implying that it is positive semidefinite.

From this point on the proof is technical and long and boils down to bounding the eigenvalues of the random matrices involved in  $M$ . For a complete analytical proof we point the reader to Section A.3 in the Appendix of [15].  $\square$

What should also be noted at this point is that the technique described is a constructive one. The planted partitions can be recovered with little effort.

### 3.3 Spectral Partitioning of Random Graphs

During the same year in parallel with the work of Feige and Kilian [15], McSherry in [30] also worked on partitioning problems on random graphs. On his seminal work he

presented a single framework that models three vertex partitioning problems GRAPH PARTITIONING,  $k$ -COLORING and CLIQUE and gave parameter range guarantees for each one of them, as well as an algorithm that achieved them.

We present the main result on GRAPH PARTITIONING from their paper without stating the proof.

**Theorem 3.3.1** ([30]). *There is a constant  $c$  such that for sufficiently large  $n$  if*

$$p - q > c\sqrt{p \frac{\log(n/\delta)}{n}} \tag{3.15}$$

*then we can recover the planted partition with probability  $1 - \delta$ .*

The range of parameters is equivalent to the range in [7], up to constant factors. Note though that this is the first actual work on MULTISECTION, both of the previous works were on BISECTION.

The techniques applied by McSherry involve a lot of algebraic graph theory and are very elegant, based on matrix perturbations and projections, though we choose not to present them in this thesis in the interest of time. We urge the inquiring reader to refer to the paper [30] for all the techniques and detailed results.

## 3.4 The Current State-of-the-Art

Ever since 2001 a lot has transpired in the area of partitioning and clustering. With the rise of data science and big data, the need to have ever more efficient algorithms to solve such primitive tasks has increased exponentially. Research in the field has expanded from mathematicians and theoreticians to scientists in statistics, machine learning, information retrieval and more. For surveys on recent advances look at [9] and the references there in.

With so much diversity in the backgrounds of the researchers involved, many new models arose during the past 10-15 years, serving the needs of different data-sets or different scenarios. But the most popular remained to be the standard - more natural one - planted partition model. So, a large body of literature [8], [33], [25], [10], [30], [31] and [12] focused on trying to improve the range of parameters  $p - q$  for which we could retrieve the partition.

The best bound still for the general balanced graph partitioning case for  $k \geq 2$  seems to be, to our knowledge, from [30] ensuring recovery for  $(p - q)/\sqrt{p} \geq \Omega\left(\sqrt{\log n/n}\right)$ , and has not been improved for more than a decade.

### 3.4.1 Sharp Thresholds for Recovery in the Planted Partition Model

More recently, a new phenomenon has been identified for the planted partition model in a regime where  $p = a/n$  and  $q = b/n$  [13]. In this regime, exact recovery is not possible, since the graph is, with high probability, not connected. However, *partial recovery* is possible, and the focus has been shifted on determining for which regime of  $a$  and  $b$  it is possible to obtain a reconstruction of the blocks which is asymptotically better than a random guess. We will refer to this reconstruction requirement as *detection*.

In [13], it was conjectured that detection is possible if and only if  $(a-b)^2 > 2(a+b)$ . This is a particularly fascinating and strong conjecture, as it provides a necessary and sufficient condition for detection with a sharp closed-form expression and it was very recently proven by [29] with information theoretic techniques.

One of the latest results of 2014 is that of Abbe *et al* [1] where they answer the question of whether it is possible to establish a similar sharp phase transition bound for *recovery* in the stochastic block model for  $k = 2$  positively. In particular they

show sharp regimes in which recovery is *feasible* and *impossible*. In particular they proved the following result.

**Theorem 3.4.1** ([1]). *Let*

$$a = p \frac{n}{\log n} \quad \text{and} \quad b = q \frac{n}{\log n} \quad (3.16)$$

*and without loss of generality assume that  $a > b \geq 0$ . If*

$$\frac{a+b}{2} - \sqrt{ab} < 1 \quad (3.17)$$

*then recovery is impossible and if*

$$\frac{a+b}{2} - \sqrt{ab} > 1 \quad (3.18)$$

*then recovery of the planted partition is possible with high probability.*

Also they propose an efficient algorithm based on a semidefinite programming relaxation of *maximum likelihood (ML)*, which is proved to succeed in recovering the blocks close to the threshold, matching the state-of-the-art asymptotic bound held by McSherry [30] for the case of  $k = 2$  blocks and improve upon the constants, while numerical experiments suggest it may achieve the threshold.

Chen and Xu in [11] working on the general case of  $k \geq 2$  observe the following phenomenon: The parameter space can be partitioned into four disjoint regions, such that each region corresponds to statistically easier instances of the problem than the previous region, and recovery can be achieved by simpler algorithms with lower running time. Significantly, there are large gaps between the statistical performance of computationally expensive algorithms and that of computationally efficient algorithms. Their main theorems identify the following four regimes of the problem defined by the values of the quantity  $\frac{(p-q)^2}{q(1-q)}$ .

- **The Impossible Regime:**

$$\frac{(p-q)^2}{q(1-q)} \lesssim \frac{k}{n} \quad (3.19)$$

In this regime, there is no algorithm regardless of its computational complexity, that can recover the clusters with reasonable probability.

- **The Hard Regime:**

$$\frac{k}{n} \lesssim \frac{(p-q)^2}{q(1-q)} \lesssim \frac{k^2}{n} \quad (3.20)$$

There exists an algorithm - specifically the exponential time Maximum Likelihood Estimator (MLE) - that recovers the blocks with high probability in this regime as well as in the next two easier regimes; we omit such implications in the sequel). There is no known polynomial-time algorithm for this regime.

- **The Easy Regime:**

$$\frac{k^2}{n} \lesssim \frac{(p-q)^2}{q(1-q)} \lesssim \frac{k}{\sqrt{n}} \quad (3.21)$$

There exists a polynomial time algorithm - specifically a convex relaxation of MLE - that recovers the blocks with high probability in this regime. Moreover, this algorithm provably fails in the hard regime above.

- **The Simple Regime:**

$$\frac{(p-q)^2}{q(1-q)} \gtrsim \frac{k}{\sqrt{n}} \quad (3.22)$$

A simple algorithm based on counting vertex degrees and common neighbors recovers the blocks with high probability in this regime, and provably fails outside this regime.

Chen and Xu conjecture that no polynomial time algorithm can succeed in the hard regime. In support of their belief they also note that the hard regime contains the

---

<sup>1</sup> The notation  $\lesssim$  and  $\gtrsim$  ignores constant and  $\log n$  factors, while the notation  $\lesssim$  and  $\gtrsim$  ignores just constant factors.

standard PLANTED CLIQUE problem with clique size  $K = o(\sqrt{n})$ , which has no polynomial-time algorithm so far despite decades of effort and is widely believed to be computationally intractable [22], [26].

They also consider a polynomial-time algorithm based on a convex relaxation of the MLE, and obtain nearly matching sufficient and necessary conditions for the success of the algorithm. It shows that the algorithm does not achieve the mini-max lower bounds. Their performance guarantee improves upon all existing ones for polynomial-time algorithms, especially when the number of clusters is large.

# Chapter 4

## A Different Spectral Approach

” *From childhood’s hour I have not been  
As others were - I have not seen  
As others saw - I could not bring  
My passions from a common spring*

”

---

Edgar Allan Poe, *Alone*

Our contribution regards the unweighted version of the GRAPH PARTITIONING problem for  $k \geq 2$ . Being inspired by the works of [7] and [15] we wanted to extend these techniques to the MULTISECTION case in hope for a better or equal bound to that of [30] with a different and easier proof, hopefully falling under the harder regime of [11].

It is worth to be noted that this work was concurrent to that of [11] and in just days before we concluded the result, we got news of their paper. In the end this technique cannot improve the state-of-the-art, but we still believe it holds some interest on its own right and that it is more elegant than other techniques. Initially, we believed



that we could potentially push our parameters to be in the “hard” regime of [11], but that does not seem possible with this spectral technique.

Notice that we can extend our model into a semi-random one by adding a monotone adversary as in [15]. Similarly to [15] our function can be shown to be *robust* for the case of a (monotone) adversary. This makes the model more robust since on top of the randomness, there is an (almost) arbitrary removal and addition of edges.

We begin with the SDP first introduced in [19] and take the standard Dual problem which for SDPs has only a single positive semidefinite constraint. We make a choice of the dual variables that will make the objective function yield the required quantity and then show that for that choice the matrix constraint can be satisfied. Furthermore, since the SDP values are integral, an algorithm construction is almost straightforward.

As [7], [15] and many others, we embrace the planted partition model as discussed in Chapter 3. For completeness we restate it here.

## 4.0.2 The Planted Partition Model

Let  $G = (V, E)$  be a graph on  $n$  vertices where  $E = \emptyset$ . Then

- Fix a partition of the vertices of  $G$  into  $V_1, \dots, V_k$  equal sized blocks,
- For every pair of vertices inside a block add an edge with independent probability  $p$ ,
- For every pair of vertices across distinct block add an edge with independent probability  $q < p$ .

## 4.1 The SDP

The SDP we are considering is:

$$\begin{aligned} \min \quad & A \bullet X \\ \text{s.t.} \quad & J \bullet X = n^2/k \\ & X_{ii} = 1 \\ & X_{ij} \geq 0 \\ & X \succcurlyeq 0 \end{aligned} \tag{4.1}$$

We rewrite  $X_{ij} \geq 0$  by introducing the slack variables  $s_{ij}$  as  $X_{ij} - s_{ij} = 0$  and get:

$$\begin{aligned} \min \quad & A \bullet X \\ \text{s.t.} \quad & J \bullet X = n^2/k \\ & X_{ii} = 1 \\ & X_{ij} - s_{ij} = 0 \\ & X \succcurlyeq 0 \end{aligned} \tag{4.2}$$

## 4.2 The Dual

As per standard duality theory we write the *Dual* of the above SDP:

$$\begin{aligned} \max \quad & \sum_{i=1}^n y_i + (n^2/k)y_0 \\ \text{s.t.} \quad & Y - A + y_0 J - Z \succcurlyeq 0 \end{aligned} \tag{4.3}$$

Where  $Y$  is the diagonal  $y_i$  variable matrix,  $A$  is the adjacency matrix of the graph,  $J$  is the identity matrix and  $Z$  is an auxiliary variable matrix that only affects feasibility, but does not appear in the objective function.

Recall that as in *linear programming*, similarly in semi-definite programming a solution to the dual problem provides a bound on the value of the solution to the

primal problem; when the problem is convex and satisfies the constraint qualifications, then the value of an optimal solution of the primal problem is given by the dual problem. Which in our case is true, thus it suffices to calculate the value of the dual problem.

### 4.3 The Approach

At this point it will be useful to give a couple of definitions:

**Definition 4.3.1.** Consider a  $k$ -partite graph  $G_k = (\{V_i\}_{i=1}^k, E)$  and let  $d_{out}^i(j)$  be the number of edges that go from vertex  $i$  to block  $j$ . We call the  $k$ -partite graph  $G_k$  *r-canonical* if

$$\forall(i, j) d_{out}^i(j) = r. \quad (4.4)$$

We will denote a random  $k$ -partite  $r$ -canonical graph as  $G_k^r$ .

We proceed to partition the space  $\mathbb{R}^n$  in the following perpendicular subspaces:

$$\mathbb{1} : \text{the all-ones vector}, \quad (4.5)$$

$$R_{k-1} : \text{the subspace perpendicular to } \mathbb{1} \text{ with equal values in each block}, \quad (4.6)$$

$$R_{n|k-1} : \text{the subspace where the sum on each block is equal to 0}. \quad (4.7)$$

Following are some simple facts:

**Observation 4.3.1.** *The space  $R_{k-1}$  is an eigenspace for any  $r$ -canonical graph with eigenvalue  $-r$ .*

**Observation 4.3.2.** *The space  $R_{n|k-1}$  is an eigenspace for the complete  $k$ -partite graph with eigenvalue 0.*

Now looking at the dual problem, by decomposing  $A$  to  $A_p$  and  $A_q$ , where  $A_p$  is the adjacency matrix of the subgraph with only edges within blocks and  $A_q$  the one

with just edges between distinct blocks, and substituting  $A_p$  with  $D_p - L_p$ , we can rewrite the above constraint as follows:

$$M = Y - D_p + y_0 J - Z + L_p - A_q \succcurlyeq 0. \quad (4.8)$$

So the problem now reduces to finding proper choices for the  $y_i$ 's and  $y_0$  such that we measure the total cut in the objective function and at the same time meet the dual constraint.

It will be useful to define the following variables to help unclutter the notation:

$$E_q := \frac{qn}{k}, \quad E_p := \frac{pn}{k}. \quad (4.9)$$

and

$$\Delta_q := \sqrt{\frac{qn}{k} \log n}, \quad \Delta_p := \sqrt{\frac{pn}{k} \log n}. \quad (4.10)$$

We have the following simple lemma which follows via *Chernoff* bounds.

**Lemma 4.3.3.**  $\forall(i, j)$

$$d_{out}^i(j) \leq E_q + O(\Delta_q) \quad (4.11)$$

with high probability.

*Proof.* Let  $X = d_{out}^i(j)$ . Since  $X \sim \text{Binomial}(n/k, q)$  by standard *Chernoff* bounds it holds:

$$\Pr \left[ X \geq \frac{qn}{k} + \sqrt{\log n} \cdot \sqrt{\frac{qn}{k}} \right] \leq \exp \left( -\frac{(\sqrt{\log n})^2}{2} \right) \quad (4.12)$$

$$\Pr \left[ X \geq \frac{qn}{k} + \sqrt{\frac{qn}{k} \log n} \right] \leq \frac{1}{n^2}. \quad (4.13)$$

□

Notice that  $Z$  does not appear in the objective function and so we are free to make the best choice that satisfies our constraint.

We now describe the way we choose  $Z$ .

The sampled  $A_q$  may not be canonical. Consider the set of all graphs  $\hat{Z}$  such that  $A_q \cup Z$ , where  $Z \in \hat{Z}$  is  $(E_q + c\Delta_q)$ -canonical (where  $c$  is some constant). We make this choice of  $Z$  uniformly at random from  $\hat{Z}$ . Intuitively,  $Z$  will be the graph composed of the remaining edges that would make  $A_q$   $(E_q + c\Delta_q)$ -canonical.

**Claim 4.3.4.** *For a suitable  $c$  with high probability the corresponding set  $\hat{Z}$  from  $A_q$  will be non-empty.*

We now describe the choice of dual parameters:

$$y_i = d_{in}^i - (E_q + c\Delta_q) \quad (4.14)$$

and

$$y_0 = (E_q + c\Delta_q)(k/n). \quad (4.15)$$

Notice firstly that for the above choice of dual parameters the dual objective function gives exactly the maximum sum of the degrees inside the blocks, which means it maximizes the number of edges inside the blocks, which in turn means that we are minimizing the number of edges across blocks, which is exactly what we wanted to do.

Secondly, under the selected parameters described above we see that  $Z + A_q$  is now a  $(E_q + c\Delta_q)$ -canonical graph as we wanted.

Using the properties of canonical graphs we get the following easily derivable lemmata.

**Lemma 4.3.5.**  $\mathbb{1}$  is an eigenvector of  $M$  with eigenvalue  $= 0$ .

*Proof.* We show that  $x^T Mx = 0$ , for  $x = \mathbb{1}$ .

$$x^T Mx = x^T(Y - D_p + y_0J + L_p - Z - A_q)x \quad (4.16)$$

$$= x^T Yx - x^T D_p x + x^T(y_0J)x + x^T L_p x - x^T Zx - x^T A_q x \quad (4.17)$$

$$= x^T(Y - D_p)x + n^2 y_0 - x^T(Z + A_q)x \quad (4.18)$$

$$= (E_q + c\Delta_q)(k - 1) - (k - 1)(E_q + c\Delta_q) \quad (4.19)$$

$$= 0 \quad (4.20)$$

□

**Lemma 4.3.6.**  $R_{k-1}$  is an eigenspace of  $M$  with eigenvalue  $= 0$ .

*Proof.* We show that  $y^T My = 0$ , for  $y \in R_{k-1}$ .

$$y^T My = y^T(Y - D_p + y_0J + L_p - Z - A_q)y \quad (4.21)$$

$$= y^T Yy - y^T D_p y + y^T(y_0J)y + y^T L_p y - y^T Zy - y^T A_q y \quad (4.22)$$

$$= y^T(Y - D_p)y - y^T(Z + A_q)y \quad (\text{Observation 4.3.1}) \quad (4.23)$$

$$= (E_q + c\Delta_q) - (E_q + c\Delta_q) \quad (4.24)$$

$$= 0 \quad (4.25)$$

□

The above lemmata imply that it suffices show that  $x^T Mx \geq 0$ , for all  $x \in R_{n|k-1}$ .

To this end we first need to understand the behavior of  $Z + A_q$  on this space. We first make the following observation:

**Observation 4.3.7.** Let  $\hat{A}_q$  be the distribution conditioned on the event that the corresponding set  $\hat{Z}$  is non-empty. Then  $Z + \hat{A}_q$  is the same distribution as a random  $(E_q + c\Delta_q)$ -canonical graph  $G_{E_q + c\Delta_q}$ .

**Claim 4.3.8.** *Restricted to the space  $R_{n|k-1}$  we have that*

$$\lambda(G_r) \leq \sqrt{rk \log n} \quad (4.26)$$

**Observation 4.3.9.** *It is also true that*

$$\lambda(L_p) \geq E_p - \Delta_p \quad (4.27)$$

Using the above lemmata and everything that we have stated thus far we get the following corollary.

**Corollary 4.3.10.** *The SDP recovers the original solution with high probability when*

$$p - q \geq O \left( \sqrt{\frac{kp \log n}{n}} + k \sqrt{\frac{q \log n}{n}} \right) \quad (4.28)$$

*Proof.* Recall that for our choice of the  $y_i$ 's,  $y_0$  and the matrix  $Z$  the objective function of the dual formulation 4.3 calculates exactly the quantity we want. Hence, the only thing left to show is that the constraint is met.

Suffice to show that  $x^T M x \geq 0$ , for all  $x \in \mathbb{R}^n$ . In particular by the subspace decomposition of  $\mathbb{R}^n$  that we showed already it is enough to show that:

$$x^T M x = 0, x = \mathbb{1} \quad (4.29)$$

$$y^T M y = 0, y \in R_{k-1} \quad (4.30)$$

$$z^T M z \geq 0, z \in R_{n|k-1} \quad (4.31)$$

By Lemma 4.3.5, we satisfy (4.29) and by Lemma 4.3.6, we satisfy (4.30) also. Suffices to show (4.31).

$$z^T M z \geq 0 \iff z^T (Y - D_p + y_0 J + L_p - Z - A_q) z \geq 0 \quad (4.32)$$

$$\iff z^T Y z - z^T D_p z + z^T (y_0 J) z + z^T L_p z - z^T Z z - z^T A_q z \geq 0. \quad (4.33)$$

By definition  $z^T (Y - D_p) z = \{y_i - d_{in}^i\}_{i=1}^n$ , which by (4.14) becomes just  $-(E_q + c\Delta_q)$ .

$$-(E_q + c\Delta_q) + z^T (y_0 J) z + z^T L_p z - z^T Z z - z^T A_q z \geq 0. \quad (4.34)$$

The identity matrix  $z^T (y_0 J) z$  goes to 0 for this choice of  $z$ .

$$-E_q - c\Delta_q + z^T L_p z - z^T Z z - z^T A_q z \geq 0. \quad (4.35)$$

By Observation 4.3.9 we get:

$$-E_q - c\Delta_q + E_p - \Delta_p - z^T Z z - z^T A_q z \geq 0. \quad (4.36)$$

Finally, by Lemma 4.3.8 we get:

$$-E_q - c\Delta_q + E_p - \Delta_p - \sqrt{(E_q + c\Delta_q)k \log n} \geq 0. \quad (4.37)$$

Substituting back the  $E_q, E_p, \Delta_q, \Delta_p$  we get:

$$p - q \geq O \left( \sqrt{\frac{kp \log n}{n}} + k \sqrt{\frac{q \log n}{n}} \right). \quad (4.38)$$

As promised. □

It just caught our attention that the proof of Claim 4.3.8 has an error and sadly we will have to leave it unproved at this time.



# Chapter 5

## Open Problems and Future Work

” *The best thing about the future is that it comes one day at a time.* ”

---

Abraham Lincoln, 1809 - 1865

It is an interesting question to what extent the multitude of results sketched above have reached a state of maturity where future improvements become less and less likely. On the other hand, as soon as you widen your view in some direction, there are plenty of important open problems.

As we said in the end of Chapter 3, for the case of bipartitioning [1] not only provided a sharp information theoretic threshold characterizing the feasibility of recovery in the planted partition - stochastic block model but also gave an algorithm that asymptotically matches the previous state-of-the-art given in [30], which in absolute-constant terms even improves it, answering most of standard open questions on the case of two blocks.

One open problem left by [1] is finding a tight analysis, as their algorithm is not in theory achieving exactly the threshold, though in numerical simulations it does, for their algorithm or some other spectral algorithm. Additionally it would be interesting

to see if some other, new algorithm could be suggested that achieves the same bound with completely different techniques.

In the area of thresholds for exact recovery in the case of arbitrary number of blocks; the general balanced Graph Partitioning problem, several future directions are of interest. It would be useful in practice to allow for a finer spectrum, ideally close to a continuum, of computational-statistical tradeoffs.

Moreover, what is still an important open problem in the case of arbitrary blocks  $k > 2$  is whether there exists an algorithm in the  $p - q$  parameter range regime characterized as “hard” in [1]. If such an algorithm existed it could have interesting implications on the notorious PLANTED CLIQUE problem.

Finally, it would also be interesting to study extensions to the setting with overlapping clusters, and to the case where the values of the model parameters are unknown. Establishing conditional computational hardness of the support estimation version of the planted clustering problem is also an interesting problem.

# Bibliography

- [1] Emmanuel Abbe, Afonso S. Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *CoRR*, abs/1405.3267, 2014.
- [2] N. Alon. Eigenvalues and expanders, 1986.
- [3] Konstantin Andreev and Harald Räcke. Balanced graph partitioning. In *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '04, pages 120–124, New York, NY, USA, 2004. ACM.
- [4] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 284–293, New York, NY, USA, 1995. ACM.
- [5] Sanjeev Arora, Subhash A. Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: Extended abstract. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 21–28, New York, NY, USA, 2008. ACM.
- [6] David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, editors. *Graph Partitioning and Graph Clustering - 10th DIMACS Implementing - 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, volume 588 of *Contemporary Mathematics*. American Mathematical Society, 2013.
- [7] Ravi B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, SFCS '87, pages 280–285, Washington, DC, USA, 1987. IEEE Computer Society.
- [8] Thang Nguyen Bui, Soma Chaudhuri, Frank Thomson Leighton, and Michael Sipser. Graph bisection algorithms with good average case behavior. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 181–192, 1984.
- [9] Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. *CoRR*, abs/1311.3144, 2013.

- [10] Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 903–909, 2001.
- [11] Y. Chen and J. Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *ArXiv e-prints*, feb 2014.
- [12] David S. Choi, Patrick J. Wolfe, and Edoardo M. Airoldi. Stochastic blockmodels with growing number of classes. *CoRR*, abs/1011.4644, 2010.
- [13] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *CoRR*, abs/1109.3041, 2011.
- [14] Guy Even, Joseph (Seffi) Naor, Satish Rao, and Baruch Schieber. Fast approximate graph partitioning algorithms. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '97*, pages 639–648, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.
- [15] Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *J. Comput. Syst. Sci.*, 63(4):639–671, 2001.
- [16] Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, 31(4):1090–1118, April 2002.
- [17] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.
- [18] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [19] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.
- [20] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2009.
- [21] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
- [22] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best nash equilibrium? *SIAM J. Comput.*, 40(1):79–91, 2011.

- [23] Bruce Hendrickson and Tamara G. Kolda. Graph partitioning models for parallel computing. *Parallel Comput.*, 26(12):1519–1534, November 2000.
- [24] L[aurent] Hyafil and R[onald] L. Rivest. Graph partitioning and constructing optimal decision trees are polynomial complete problems. Technical Report Rapport de Recherche no. 33, IRIA – Laboratoire de Recherche en Informatique et Automatique.
- [25] Mark Jerrum and Gregory B. Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(1-3):155–175, 1998.
- [26] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Des. Codes Cryptography*, 20(3):269–280, 2000.
- [27] Andrew B. Kahng, Jens Lienig, Igor L. Markov, and Jin Hu. *VLSI Physical Design - From Graph Partitioning to Timing Closure*. Springer, 2011.
- [28] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC '14*, pages 41–49, New York, NY, USA, 2014. ACM.
- [29] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 694–703, 2014.
- [30] F. McSherry. Spectral partitioning of random graphs. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science, FOCS '01*, pages 529–, Washington, DC, USA, 2001. IEEE Computer Society.
- [31] Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *CoRR*, abs/1311.4115, 2013.
- [32] Horst D. Simon and Shang-Hua Teng. How good is recursive bisection? *SIAM J. Sci. Comput.*, 18(5):1436–1445, September 1997.
- [33] Tom A.B. Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.