Semi-random Graphs with Planted Sparse Vertex Cuts: Algorithms for Exact and Approximate Recovery

Anand Louis
Indian Institute of Science
Bangalore, India
anandl@iisc.ac.in

Rakesh Venkat[1]
Hebrew University of Jerusalem
Israel
rakesh@cs.huji.ac.il

Abstract

The problem of computing the vertex expansion of a graph is an NP-hard problem. The current best worst-case approximation guarantees for computing the vertex expansion of a graph are a $\mathcal{O}\left(\sqrt{\log n}\right)$-approximation algorithm due to Feige et al. [16], and $\mathcal{O}\left(\sqrt{\mathsf{OPT}\log d}\right)$ bound in graphs having vertex degrees at most d due to Louis et al. [29].

We study a natural semi-random model of graphs with sparse vertex cuts. For certain ranges of parameters, we give an algorithm to recover the planted sparse vertex cut exactly. For a larger range of parameters, we give a constant factor bi-criteria approximation algorithm to compute the graph's balanced vertex expansion. Our algorithms are based on studying a semidefinite programming relaxation for the balanced vertex expansion of the graph.

In addition to being a family of instances that will help us to better understand the complexity of the computation of vertex expansion, our model can also be used in the study of community detection where only a few nodes from each community interact with nodes from other communities. There has been a lot of work on studying random and semi-random graphs with planted sparse edge cuts. To the best of our knowledge, our model of semi-random graphs with planted sparse vertex cuts has not been studied before.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Semi-Random models, Vertex Expansion, Approximation Algorithms, Beyond Worst Case Analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.101

Related Version A full version of the paper is available at `https://arxiv.org/abs/1805.09747`.

Acknowledgements We thank Amit Deshpande for many helpful discussions. Rakesh Venkat was visiting Microsoft Research, Bangalore when this research was initiated. Anand Louis is grateful to MSR, Bangalore for supporting this collaboration.

[1] Supported by an I-CORE Algorithms Fellowship

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella;
Article No. 101; pp. 101:1–101:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Given a graph $G = (V, E)$, the vertex expansion of a non-empty subset $S \subset V$, denoted by $\phi^{\mathsf{V}}(S)$, is defined as[2]

$$\phi^{\mathsf{V}}(S) \stackrel{\text{def}}{=} |V| \, \frac{|N(S)| + |N(V \setminus S)|}{|S| \, |V \setminus S|} \,,$$

where $N(S)$, the neighborhood of S, is defined as $N(S) \stackrel{\text{def}}{=} \{j \in V \setminus S : \exists i \in S$ such that $\{i, j\} \in E\}$. The vertex expansion of the graph G, denoted by ϕ^{V}_G, is defined as $\phi^{\mathsf{V}}_G \stackrel{\text{def}}{=} \min_{S \subset V,\ S \neq \emptyset} \phi^{\mathsf{V}}(S)$. Computing the vertex expansion of a graph is NP-hard. The complexity of computing various graph expansion parameters are central open problems in theoretical computer science, and inspite of many decades of intensive research, they are yet to be fully understood [8, 9, 26, 10, 16, 39].

Feige et al. [16] gave a $\mathcal{O}\left(\sqrt{\log n}\right)$-approximation algorithm for computing the vertex expansion of a graph. Louis et al. [29] gave an algorithm that computes a set having vertex expansion at most $\mathcal{O}\left(\sqrt{\phi^{\mathsf{V}} \log d}\right)$ in graphs having vertex degrees at most d. We give a brief description of other related works in Section 1.3. In this work, we study a natural semi-random family of graphs, and give polynomial time exact and approximation algorithms for computing the *balanced vertex expansion* (a notion that is closely related to the vertex expansion of a graph, we define it formally in Section 1.1) w.h.p.

In many problems, there is a huge gap betwen theory and practice; the best known algorithms provide a somewhat underwhelming performance guarantee, however simple heuristics perform remarkably well in practice. Examples of this include the simplex algorithm for linear programming [25], SAT [11], sparsest cut [22, 23], among others. In many cases, the underwhelming provable approximation guarantee of an algorithm is a property (hardness of approximation) of the problem itself; even in many such cases, simple heuristics work remarkably well in practice. A possible explanation for this phenomenon could be that for many problems, the instances arising in practice tend to have some inherent structure that makes them "easier" than the worst case instances. Many attempts have been made to understand the structural properties of these instances, and to use them in designing algorithms specifically for such instances, which could perform much better than algorithms for general instances. A fruitful direction of study has been that of modelling real world instances as a family of random and semi-random instances satisfying certain properties. Our work can be viewed as the study of the computation of vertex expansion along this direction.

Often graphs with sparse cuts are used to model communities. For example, the vertices of a graph can be used to represent the members of the communities, and two vertices would have an edge between them if the members corresponding to them are *related* in some way. In such a graph, the sparse cuts indicate the presence of a small number of relations across the members corresponding to the cut, which are likely to be some form of communities within the members. The *stochastic block models* have been used to model such communities. Our model can also be viewed as model for communities where only a few members from each community have a relationship with members from another community.

[2] Other definitions of vertex expansion have been studied in the literature, see Section 1.3.

1.1 Vertex Expansion Block Models

For a graph $G = (V, E)$, its *balanced vertex expansion* $\phi^{\mathsf{V-bal}}$ is defined as

$$\phi_G^{\mathsf{V-bal}} \stackrel{\text{def}}{=} \min_{\substack{S \subset V \\ |S| = |V|/2}} \phi^{\mathsf{V}}(S) \,.$$

Another common notion of vertex expansion that has been studied in the literature is $\phi^{\mathsf{V,a}}(S) \stackrel{\text{def}}{=} (|V|\,|N(S)|\,/\,(|S|\,|V \setminus S|))$, and as before, $\phi_G^{\mathsf{V,a}} \stackrel{\text{def}}{=} \min_{S \subset V} \phi^{\mathsf{V,a}}(S)$. [29] showed that the computation ϕ_G^{V} and $\phi_G^{\mathsf{V,a}}$ is equivalent upto constant factors. In this work, we develop a semi-random model for investigating the balanced vertex expansion of graphs.

We study instances that are constructed as follows. We start with a set of n vertices, and we arbitrarily partition them into two sets S, S' of $n/2$ vertices each. Next, we choose a small subset $T \subset S$ of size εn (resp. $T' \subset S'$) to form the vertex boundary of these sets. On $S \setminus T$ (resp. $S' \setminus T'$), we add an arbitrary graph whose spectral gap[3] is at least λ (a parameter in this model), and whose vertices have roughly the same degree. We add an arbitrary low degree bipartite graph between T and T'. Between each pair of vertices in $(S \setminus T) \times T$, we add edges independently at random with probability p; this is the only part of the construction that is random. Next, we allow a *monotone adversary* to alter the graph : the monotone adversary can arbitrarily add edges that do not change the sparsity of the vertex cut (S, S'), i.e., add edges between any pair of vertices in S (resp. S'), and between any pair in $T \times T'$.

In our model, we allow the sets S and S' to be generated using different sets of parameters, i.e., we use $\varepsilon_1, \lambda_1, p_1$ for S and $\varepsilon_2, \lambda_2, p_2$ for S'. We formally define the vertex expansion block model below (see also Figure 1).

▶ **Definition 1.** An instance of $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$ is generated as follows.

1. Let V be a set of n vertices. Partition V into two sets S and S' of $n/2$ vertices each. Partition S into two sets T and $S \setminus T$ of sizes $\varepsilon_1 n$ and $(1/2 - \varepsilon_1)n$ respectively. Similarly, partition S' into two sets T' and $S' \setminus T'$ of sizes $\varepsilon_2 n$ and $(1/2 - \varepsilon_2)n$ respectively.
2. Between each pair in $(S \setminus T) \times T$ (resp. $(S' \setminus T') \times T'$), add an edge independently with probability p_1 (resp. p_2).
3. Between pairs of vertices in $S \setminus T$ (resp. $S' \setminus T'$), add edges to form an arbitrary roughly regular (formally, ratio of the maximum vertex degree and the minimum vertex degree is at most r) of spectral gap[3] at least λ.
4. Between pairs in $T \times T'$, add edges to form an arbitrary bipartite graph of vertex degrees in the range $[1, c]$ (this bipartite graph need not be connected); if $c < 1$, then add no edges in this step. We will use $\mathcal{F}$ to denote this bipartite graph.
5. (*Monotone Adversary*) Arbitrarily add edges between any pair of vertices in S (resp. S'). Arbitrarily add edges between any pair in $T \times T'$.

Output the resulting graph G.

We note that the direct analogue for vertex expansion of *Stochastic Block Models* (see related work in Section 1.3) in the regimes allowing for exact recovery is included in this setting: there, the graphs within S and S' are completely random, and so are the connections between T and T' (before the monotone adversary acts). Our model allows for a lot more adversarial action, while restricting the randomness to only a small portion of the graph.

[3] The spectral gap of a graph is defined as the second smallest eigenvalue of its normalized Laplacian matrix, see Section 1.6 for definition

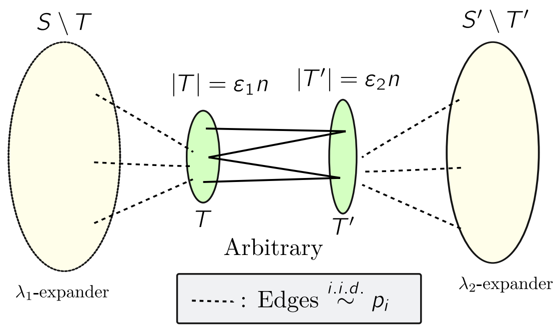

Figure 1 $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$. A monotone adversary may further add arbitrary edges within S, S' and between T, T'.

In addition to being a family of instances that will help us to better understand the complexity of the computation of vertex expansion, the vertex expansion block model can also be used in the study of community detection. In the case of two communities, the vertices in S and S' can model the members of the communities. Each community can have a few representatives who interact with the representatives from other communities; these representatives can be modelled using T and T', and their interactions can be modelled by the arbitrary graphs within T and T', and the low degree bipartite graph and the action of the monotone adversary between T and T'. Even though the connections within a community may be arbitrary, usually the members within the community are well connected with each other; this can be modelled by the choosing an appropriate values of λ_1, λ_2 plus the action of the monotone adversary. We can model the connections between community members and their representatives by a sparse random bipartite graph; our model allows the flexibility of choosing p_1 and p_2, and also the action of the monotone adversary.

1.2 Our Results

Our main result is a polynomial time algorithm for exactly recovering S and S' from a graph sampled from $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$ for certain ranges of parameters.

▶ **Theorem 2.** *There exist universal constants $c_1 \in \mathbb{R}^+, c_2 \in (0, 1/2), c_3 \in \mathbb{Z}^+, c_4 \in \mathbb{R}^+, c_5 \in (0, 1)$ satisfying the following: there exists a polynomial time algorithm which takes a graph generated from $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$, where $p_1, p_2 \in (c_1(\log n)/n, 1]$, $\varepsilon_1, \varepsilon_2 \in [1/n, c_2]$, $c \leqslant c_3$, $r \leqslant c_4$ and $\lambda_1, \lambda_2 \geqslant c_5$, and outputs the sets S and S' with probabilty at least $1 - 1/\mathsf{poly}(n)$.*

We give a description of the steps in proving Theorem 2 in Section 2; in the full version of this paper, we also observe that our proof actually shows a slightly more general result.

We also show that if the instances satisfy a few weaker requirements, then we can obtain a constant factor bi-criteria approximation algorithm for computing the balanced vertex expansion.

We study the case when $S \setminus T$ is an arbitrary graph, i.e., it does not have constant spectral gap. Note that this case is captured by setting $\lambda_1 = 0$ in our model, since the monotone adversary can create any arbitrary graph on $S \setminus T$. Our proof also allows us to let the graph induced on S' be an arbitrary graph. Again, this is captured by setting $p_2 = \lambda_2 = 0$ in our model, since the monotone adversary can create any arbitrary graph on S'. We show that we can use the underlying random bipartite graph between $S \setminus T$ and T to obtain a bi-criteria approximation algorithm in this case, that outputs an almost balanced cut with vertex expansion within a constant factor of the planted one.

▶ **Theorem 3.** *There exist universal constants $c_1 \in \mathbb{R}^+$, $c_2 \in (0, 1/2), c_6 \in (0, 1/2)$ satisfying the following: there exists a polynomial time algorithm which takes a graph generated from $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, 0, 0, 0, 0, 0)$, where $\varepsilon_1, \varepsilon_2 \in [1/n, c_2]$ and $p_1(\varepsilon_1+\varepsilon_2)n \geqslant c_1 \log n$, and outputs with probability at least $1 - 1/\mathsf{poly}(n)$, a set $A \subset V$ satisfying $|A| \in [c_6 n, (1 - c_6)n]$ and $\phi^{\mathsf{V}}(A) \leqslant \mathcal{O}(\varepsilon_1 + \varepsilon_2)$.*

Next, we study the case where the edges between $S \setminus T$ and T are arbitrary, but λ_1 is large. As in the previous case, our proof allows the graph induced on S' to be an arbitrary graph. Again, as before, this case is captured by setting $p_1 = p_2 = \lambda_2 = 0$. In this case, we show that for certain ranges of λ_1, we can obtain a constant factor bi-criteria approximation algorithm for computing the balanced vertex expansion.

▶ **Theorem 4.** *There exist universal constants $c_2 \in (0, 1/2), c_5 \in (0, 1), c_6 \in (0, 1/2)$ satisfying the following: there exists a polynomial time algorithm which takes a graph generated from $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, 0, 0, 0, r, \lambda_1, 0)$, where $\varepsilon_1, \varepsilon_2 \in [1/n, c_2]$, and $\lambda_1 \geqslant c_5 r^3(\varepsilon_1 + \varepsilon_2)$, and outputs a set $A \subset V$ satisfying $|A| \in [c_6 n, (1 - c_6)n]$ and $\phi^{\mathsf{V}}(A) \leqslant \mathcal{O}(\varepsilon_1 + \varepsilon_2)$.*

In fact, we prove a stronger result: it suffices for $S \setminus T$ to contain a subgraph on $\Omega(n)$ vertices having spectral gap at least $c_5 r^3(\varepsilon_1 + \varepsilon_2)$, to obtain a constant factor bi-criteria approximation algorithm for computing the balanced vertex expansion.

Organization of the rest of the paper

We will next describe related work in Section 1.3, and give our SDP relaxation in Section 1.4. We give an overview of our proofs in Section 1.5, and present a slightly more detailed description of the proof of our main result, Theorem 2, in Section 2. For the proofs of Theorem 3 and Theorem 4, we refer the reader to the full version of the paper.

1.3 Related Work

Stochastic Block Models

Closely related to the vertex expansion of a graph is the notion of edge expansion which is defined as follows.

▶ **Definition 5.** For a weighted graph $G = (V, E, w)$, with non-negative edge weights $w : E \to \mathbb{Q}^+$, the edge expansion of a non-empty set $S \subset V$ is defined as

$$\phi_G(S) \stackrel{\text{def}}{=} \frac{\sum_{e \in E(S, V \setminus S)} w(e)}{\min\{\mathsf{vol}(S), \mathsf{vol}(V \setminus S)\}},$$

where $E(S, V \setminus S) \stackrel{\text{def}}{=} \{\{i, j\} \in E : i \in S, j \notin S\}$ and $\mathsf{vol}(S) \stackrel{\text{def}}{=} \sum_{i \in S} \sum_{j \sim i} w(\{i, j\})$. The edge expansion of the graph is defined as $\phi_G \stackrel{\text{def}}{=} \min_{S \subset V,\ S \neq \emptyset} \phi_G(S)$.

The *Stochastic Block Model* (we will refer to it as the *edge expansion stochastic block model* to differentiate it from our block model) is a randomized model for instances that are generated as follows. A set of n vertices is arbitrarily partitioned into sets S, S' of equal sizes. Between each pair of vertices in S, an edge is added independently with probability p, and between each pair of vertices in $S \times S'$, an edge is added independently with probability q (typically $p > q$).

Starting with work of Holland et al.[20], the works of Boppana [13], who gave a spectral algorithm, and of Jerrum and Sorkin [21], who gave a metropolis algorithm, contributed significantly to the study of stochastic block models. One of the break through works in the study of SBMs is the work of McSherry [34], who gave a simple spectral algorithm for a certain range of parameters. There has been a lot of recent work related to a certain conjecture regarding SBMs, which stated the regime of parameters p, q for which it was possible to detect the presence of communities. The works [36, 37, 38, 33] have contributed to proving various aspects of the conjecture. In a recent work, Abbe et al.[3] showed that the natural SDP relaxation for balanced edge expansion is integral when there is a sufficient gap between p and q, and $p, q = \Omega(\log n/n)$; Mossel et al.[38] gave an algorithm for a larger regime of parameters which was not based on semidefinite programming. More general SBMs have been studied by Abbe and Sandon [4, 5, 6], Aggarwal et al.[7], etc.

Kim et al.[24] studied a version of SBM for hypergraphs, and gave algorithms for it based on studying a certain "adjacency tensor", the analog of the adjacency matrix for hypergraphs. They also study the sum-of-squares algorithms for this model. [27] gave a reduction from vertex expansion problems to hypergraph expansion problems. We note that applying this reduction to the instances from our models does not give the model studied by [24]: this reduction will only introduce hyperedges between the sets corresponding to T and T', whereas the model studied by [24] adds random hyperedges between S and S'. Moreover, many parts of the graph from our model are adversarially chosen.

Semi-random models for edge expansion problems

Monotone adversarial errors in SBMs are the arbitrary addition of edges between pairs of vertices within S (resp. S'), and the arbitrary deletion of existing edges between S and S'. Feige and Kilian [17] gave an algorithm for the edge expansion model with monotone adversarial errors when the gap between p and q is sufficiently large. Guedon and Vershynin [18] gave an algorithm based on semidefinite programming for partially recovering the communities for certain ranges of parameters. Moitra et al.[35] gave algorithms (based on semidefinite programming) and lower bounds for partial recovery in the stochastic block model with a monotone adversary. Makarychev et al.[32] gave an algorithm for partial recovery for the stochastic block model with a monotone adversarial errors and a small number of arbitrary errors (i.e. non-monotone errors).

Makarychev et al.[30, 31] studied some semi-random models of instances for edge expansion problems. In particular, [30] studied a model analogous to $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, 0, 0, 0, r, \lambda_1, 0)$for edge expansion problems; they showed that if the number of edges crossing (S, S') is εm, and if there is a set of m edges E_1 such that (S, E_1) is a regular graph having spectral expansion at least $\Omega(\varepsilon)$, then there is an algorithm to recover a balanced cut of edge expansion $\mathcal{O}(\varepsilon)$. The proof of Theorem 4 and that of the corresponding result in [30] both proceed by using the expansion of the underlying subgraph to show that an $\Omega(n)$ sized subset of the SDP vectors lie in a ball of small radius. [30] use this to recover a constant factor bi-criteria approximation to balanced edge expansion; we adapt this approach to vertex expansion to prove Theorem 4.

The results cited here are only a small sample of the work on the SBMs. Since our model is very different from the edge expansion stochastic block models, we only give a brief survey of the literature here, and we refer the reader to a survey by Abbe [1] for a comprehensive discussion. In general, algorithms for edge expansion problems can not be used for our vertex expansion block model since sparse edge cuts and sparse vertex cuts can be uncorrelated. In particular, the action of the monotone adversary in VBM rules out the use of edge-expansion based algorithms for detecting S and S'. In the full version of this paper, we describe an explicit family graphs generated in the VBM model where these algorithms fail.

Vertex Expansion

There has been some work in investigating vertex expansion (balanced and non-balanced) the worst-case setting. Bobkov et al. [12] gave a Cheeger-type inequality for vertex expansion, where a parameter λ_∞ plays a role analogous to the use of the second eigenvalue λ_2 in Cheeger's inequality for edge expansion. Feige et al.[16] gave a $\mathcal{O}\left(\sqrt{\log n}\right)$-approximation algorithm for the problem of computing the vertex expansion of graphs. Louis et al.[29] gave an SDP rounding based algorithm that computes a set having vertex expansion at most $\mathcal{O}\left(\sqrt{\phi_G^{\mathsf{V}} \log d}\right)$, where d is the maximum vertex degree; they also showed a matching hardness result based on the Small-set expansion hypothesis. Louis and Makarychev [27] gave a bi-criteria approximation for Small-set vertex expansion, a problem related to vertex expansion. Chan et al. [14] studied various parameters related to hypergraphs, including parameters related to hypergraph expansion; they showed that many of their results extend to the corresponding vertex expansion analogues on graphs

Louis and Raghavendra [28] studied a model of instances for vertex expansion similar to ours. In their model, the adversary partitions the vertex set V into two equal sized sets S, S', and chooses a subset T (resp. T') of S (resp. S') of size at most εn. Next, the adversary chooses an arbitrary subset of pairs of vertices in S (resp. S') to form edges such that graph induced on S (resp. S') is an edge expander. The adversary chooses an arbitrary subset of the pairs of vertices in $T \times T'$ to form edges. [28] give an SDP rounding based algorithm to compute a set having vertex expansion $\mathcal{O}\left(\sqrt{\varepsilon}\right)$.

1.4 SDP Relaxation

We use the SDP relaxation for $\phi_G^{\mathsf{V-bal}}$ (SDP 6), this SDP is very similar to that of [29]. We give the dual of this SDP in SDP 7.

▶ **SDP 6** (Primal).

$$\min \sum_{i \in V} \eta_i$$

subject to

$$\begin{aligned} U_{ii} + U_{jj} - 2U_{ij} &\leqslant \eta_i & \forall i \in V, j \in N(i) \\ U_{ii} &= 1 & \forall i \in V \\ \sum_{i \in V} \sum_{j \in V} U_{ij} &= 0 & \\ U &\succeq 0 & \end{aligned}$$

▶ **SDP 7** (Dual).

$$\max \sum_{i \in V} B_{ii}$$

subject to

$$\begin{aligned} \sum_{j \in N(i)} Y_{ij} &= 1 & \\ Y_{ij} &\geqslant 0 & \forall \{i, j\} \in E \\ B_{ij} &= 0 & \forall i, j \in V, i \neq j \\ L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B &\succeq 0 & \end{aligned}$$

Here $\mathbb{1}$ denotes the all-ones vector, and $L(Y)$ denotes the Laplacian matrix of graph weight by the matrix $Y + Y^T$, i.e.

$$L(Y)_{ij} = \begin{cases} \sum_{l \in N(i)} (Y_{il} + Y_{li}) & i = j \\ -(Y_{ij} + Y_{ji}) & j \in N(i) \\ 0 & \text{otherwise} \end{cases}.$$

First, let us see why SDP 6 is a relaxation for $\phi^{\mathsf{V-bal}}$. Let P be the set corresponding to $\phi_G^{\mathsf{V-bal}}$, and let $\mathbb{1}_P \in \{-1, 1\}^n$ be a vector such that $\mathbb{1}_P(i)$ is equal to 1 if $i \in P$ and -1 otherwise. Note that since $|P| = |V|/2$, we have $\mathbb{1}^T \mathbb{1}_P = 0$. It is easy to verify that

$U := \mathbb{1}_P \mathbb{1}_P^T$ and $\eta_i := \max_{j \in N(i)} \left(\mathbb{1}_P(i) - \mathbb{1}_P(j)\right)^2$ is a feasible solution for SDP 6, and that $\sum_{i \in V} \eta_i = 4\left(|N(P)| + |N(V \setminus P)|\right)$. Therefore, $\phi_G^{\mathsf{V-bal}} = 4\left(\sum_{i \in V} \eta_i\right)/n$. and therefore, SDP 6 is a relaxation for $\phi^{\mathsf{V-bal}}$. Henceforth, we will use $\mathbb{1}_P$ to be the ± 1-indicator vector of a set $P \subset V$, i.e., $\mathbb{1}_P(i)$ is equal to 1 if $i \in P$ and -1 otherwise. We prove the following theorem about SDP 6.

▶ **Theorem 8.** *For the regime of parameters stated in Theorem 2, $U' \stackrel{\text{def}}{=} \mathbb{1}_S \mathbb{1}_S^T$ and for each i, $\eta_i' \stackrel{\text{def}}{=} \max_{j \in N(i)} \left(\mathbb{1}_S(i) - \mathbb{1}_S(j)\right)^2$ for the set S defined in $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$, is the unique optimal solution to SDP 6 with probabilty at least $1 - 1/\mathsf{poly}(n)$.*

Theorem 8 gives an algorithm to compute the matrix $\mathbb{1}_S \mathbb{1}_S^T$. By factorizing this matrix, one can obtain the vector $\mathbb{1}_S$, using which the set S can be computed. Therefore, Theorem 8 implies Theorem 2.

1.5 Proof Overview

1.5.1 Theorem 2

It is easy to verify that (U', η_i') is a feasible solution to SDP 6. Our goal will be to construct a dual solution (i.e. a feasible solution to SDP 7) which satisfies two properties,

1. The cost of this solution should be same as the cost of this primal solution (U', η').
2. The matrix $\left(L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B\right)$ should have rank $n-1$.

Using strong duality, (1) will suffice to ensure that (U', η') is an optimal solution of the primal SDP . To show that this is the unique primal optimal solution, we will use the complementary slackness conditions which state that

$$U \cdot \left(L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B\right) = 0\,. \tag{1}$$

Since, $\left(L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B\right)$ will have rank $n-1$, this will imply that all primal optimal solutions must have rank at most 1, or in other words, there is a unique primal optimal solution (see Lemma 11).

While the approach of using complementary slackness conditions for proving the integrality of the SDP relaxation has been studied for similar problems before ([15, 2, 3, 19, 7]), there is no known generic way of implementing this approach to any given problem. Usually the challenging part in implementing this approach is in constructing an appropriate dual solution, and that, like in most of the works cited above, forms the core of our proof.

We give an outline of how we construct our dual solutions. We begin by setting the Y value for each edge added by the monotone adversary to 0, thus our proof can be viewed as saying that SDP 6 "ignores" all those edges. For the sake of simplicity, let us consider the case when the bipartite graph between T and T' is a c-regular graph. We set $B_{ii} := 4$ if $i \in T \cup T'$ and 0 if $i \notin T \cup T'$. Thus, if we can choose Y such that this choice of B is a feasible solution, then this will ensure that the cost of this dual solution, and the cost of the primal solution (U', η') are both equal to $4\left(\varepsilon_1 + \varepsilon_2\right)n$, thereby fullfilling our first requirement.

If U is a rank one matrix, and $\left(L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B\right)$ is a rank $n-1$ matrix, then (1) implies that $\mathbb{1}_S$ is an eigenvector of $\left(L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B\right)$ with eigenvalue 0. This fact will be extremely useful in setting the Y values for the edges in the bipartite graph between T and T' (Lemma 12). Now, we only have to choose the Y values for the edges fully contained in S (resp. S'). We first prove the following lemma which will help us to choose the Y values.

▶ **Lemma 9** (Informal statement of Lemma 10)**.** *There exists a constant c' such that it suffices to choose Y satisfying*

$$X^T\left(L(Y)\right)X \geqslant \frac{c'}{n}\left(\sum_{i\in S\setminus T}\sum_{t\in T}\left(X_i-X_t\right)^2+\sum_{i\in S'\setminus T'}\sum_{t\in T'}\left(X_i-X_t\right)^2\right) \qquad \forall X\in\mathbb{R}^n .$$

The proof of this lemma follows by carefully choosing the value of α, and by exploiting the fact that $\mathbb{1}_S$ is an eigenvector of $L(Y)+\alpha\mathbb{1}\mathbb{1}^T-B$ with eigenvalue 0. Proving the condition in Lemma 9 can be viewed as the problem of choosing capacities for the edges to support the multicommodity flow where each vertex $i\in S\setminus T$ wants to send c'/n amount of flow to each $t\in T$. This idea can work when S (resp. S') is a sufficiently dense graph, but does not work when S (resp. S') is sparse (for more details, we refer the reader to the full version). Our second idea is to use the edge expansion properties of the underlying spanning subgraph. For a d-regular edge expander $H=(V',E')$ having the second smallest normalized Laplacian eigenvalue λ, we get that $\sum_{\{ij\}\in E'}\left(X_i-X_j\right)^2 \geqslant (\lambda d/n)\sum_{ij\in E'}\left(X_i-X_j\right)^2$. Since $L(Y)$ is a Laplacian matrix, we get that

$$\begin{aligned}X^TL(Y)X = &\sum_{i,j\in S\{i,j\}\in E}\left(Y_{ij}+Y_{ji}\right)\left(X_i-X_j\right)^2\\ &+\sum_{i,j\in S'\{i,j\}\in E}\left(Y_{ij}+Y_{ji}\right)\left(X_i-X_j\right)^2\\ &+\sum_{i\in T,j\in T'\{i,j\}\in E}\left(Y_{ij}+Y_{ji}\right)\left(X_i-X_j\right)^2 .\end{aligned}$$

Now, since S and S' contain an almost regular edge expander as a spanning subgraph, we can adapt the expander argument to this setting and obtain some lower bound on this quantity. This strategy can work in some special cases, but fails in general. Our proof shows that the desired lower bound in Lemma 9 can be obtained using a careful combination of these two ideas, in addition to exploiting the various properties of the random graph between $S\setminus T$ and T (resp. $S'\setminus T'$ and T').

1.5.2 Theorem 3 and Theorem 4

We first solve SDP 6 and obtain a matrix U such that $U\succeq 0$. Therefore, U can be factorized into $U=W^TW$ for some matrix W. Let $u_1,\ldots,u_n$ denote the columns of this matrix W. We give an algorithm to "round" these vectors into a set satisfying the guarantees in the theorem. As in the previous case, we show that we can "ignore" all the edges added by the monotone adversary, and only focus on the edges added in step 2, 3, 4 in Definition 1.

A well known fact for edge expander graphs having roughly equal vertex degrees is that if the value of $\left\|u_i-u_j\right\|^2$ averaged over all edges $\{i,j\}$ in the graph is small, then the value of $\left\|u_i-u_j\right\|^2$ averaged over all pairs of vertices i,j in the graph is also small. In the proof of Theorem 4, we use the expansion properties of the $\Omega(n)$ sized subset of S coupled with this fact to show that an $\Omega(n)$ sized subset of the vectors $\{u_i : i\in V\}$ must lie in a ball of small diameter; this step is similar to the corresponding step of [30]. We use this to construct an embedding of the graph onto a line, and recover a cut from this embedding using an algorithm of [29]; this step can be viewed as adapting the corresponding step of [30] to vertex expansion.

In the case when $\lambda_1=0$, we show that the lopsided random bipartite graph between $S\setminus T$ and T is an edge expander w.h.p. However, this graph is not close to being regular;

the degrees of the vertices in T would be much higher than the degrees of the vertices in $S \setminus T$. Therefore, we can not directly use the strategy employed in the previous case. But we show that we can use the fact that the measure of $S \setminus T$ under the stationary distribution of the random bipartite graph between $S \setminus T$ and T is $\Omega(1)$, and that the vertices in $S \setminus T$ have roughly equal vertex degrees, to show that $\|u_i - u_j\|^2$ averaged over all pairs of vertices $i, j \in S \setminus T$ is small. From here, we proceed as in the previous case.

1.6 Notation

We denote graphs by $G = (V, E)$, where the vertex set V is identified with $[n] := \{1, 2, \ldots n\}$. For any $S \subseteq V$, we denote the induced subgraph on S by $G[S]$. Given $i \in V$ and $T \subseteq V$, define $N_T(i) \stackrel{\text{def}}{=} \{j \in T \ : \ \{i, j\} \in E\}$, and $N(i) = N_V(i)$. We denote $\Delta_T(i) := |N_T(i)|$, and $\Delta(i) = |N(i)|$. For a subgraph $\mathcal{F}$ of G, the degree of i within $\mathcal{F}$ is correspondingly $\Delta_{\mathcal{F}}(i)$.

Given a graph $G = (V, E)$ with a weight function $w : E \to \mathbb{R}_{\geqslant 0}$ on its edges, we define the *weighted* degree of a vertex $i \in V$ as $d(i) := \sum_{j \in N(i)} w_{\{i,j\}}$. In our theorems, following SDP 7, we will be assigning directed weights (or capacities) Y_{ij} to edges $\{i, j\} \in E$, and use $L(Y)$ to denote the Laplacian of G with weights $Y_{ij} + Y_{ji}$ on the edges.

Given the normalized Laplacian $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$, the *spectral gap* of G denoted by λ, is the second-smallest eigenvalue of $\mathcal{L}$.

Typically, for a vector $X \in \mathbb{R}^n$, its i-th component is denoted by X_i, or in rare cases for clarity, by $X(i)$. As in the introduction, we use $\mathbb{1}_S$ for any $S \subseteq V$ to denote the vector in $\mathbb{R}^{|V|}$ having entries $\mathbb{1}_S(i) = 1$, if $i \in S$, and -1 otherwise.

2 Exact Recovery for VBM

2.1 A sufficient condition

In order to prove Theorem 8, we start with the following lemma, which outlines a sufficient condition for integrality of the primal optimal SDP solution. For the proof of this lemma, and all subsequent lemmas in this section, we refer the reader to the full version of the paper.

▶ **Lemma 10.** *For a* $\mathsf{VBM}(n, \varepsilon_1, \varepsilon_2, p_1, p_2, c, r, \lambda_1, \lambda_2)$ *instance, if we can find a* $Y \in \mathbb{R}^{n \times n}$ *that satisfies the linear constraints on it* ($Y_{ij} \geqslant 0$ *and* $\sum_j Y_{ij} = 1$))*, and in addition has:*
(a) $\forall X \in \mathbb{R}^n$,

$$X^T \left(L(Y)\right) X \geqslant \frac{c'}{n} \left(\sum_{i \in S \setminus T} \sum_{t \in T} (X_i - X_t)^2 + \sum_{i \in S' \setminus T'} \sum_{t \in T'} (X_i - X_t)^2 \right), \tag{2}$$

(b) *For every* $i \in T, j \in T'$*, we have* $Y_{ij} = 1/\Delta_{\mathcal{F}}(i)$ *and* $Y_{ji} = 1/\Delta_{\mathcal{F}(j)}$*,*
(c) *For every* $i \in T, j \in S \setminus T$ *and* $i \in T', j \in S \setminus T'$*, we have* $Y_{ij} = 0$
where $c' = 8c/(1 - \max\{\varepsilon_1, \varepsilon_2\})$*, then SDP 6 has* (U', η') *as defined in Theorem 8 as its unique optimal solution.*

▶ Remark. Along with the SDP 7 linear constraints on Y, the above conditions ensure that we can extend Y to a feasible dual solution (Y, B, α), that satisfies the positive-semidefiniteness constraint *and* is optimal.

We begin by noting a simple consequence of complementary slackness conditions.

▶ **Lemma 11.** *Let* $M := L(Y) + \alpha \mathbb{1}\mathbb{1}^T - B$ *be constructed using an optimal dual solution* (Y, B, α)*. The primal optimal solution is integral and unique if* $\mathbb{1}_S$ *is a unique eigenvector of* M *with eigenvalue* 0.

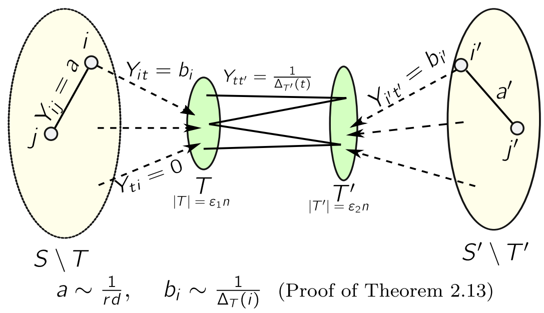

Figure 2 Outline of Y_{ij}'s used in the constructed dual solution.

It is thus sufficient to prove that the conditions in Lemma 10 imply that we can use the given Y to come up with a B and α, such that (Y, B, α) is feasible, *and* $\mathbb{1}_S$ is a unique eigenvector of M with eigenvalue 0. We first find a B (depending on Y) that yields a dual objective value of exactly $4(\varepsilon_1 + \varepsilon_2)n$ and ensures that $\mathbb{1}_S$ is an eigenvector with eigenvalue 0. We use $d_T(i) = \sum_{j \in N(i) \cap T}(Y_{ij} + Y_{ji})$ for the weighted degree of i into $T \subseteq V$.

▶ **Lemma 12.** *Fix some partial candidate dual solution (Y, α). Consider the setting of the diagonal matrix B given by:*

$$B_{ii} = \begin{cases} 2 \cdot d_{T'}(i), & \text{for } i \in T \\ 2 \cdot d_T(i), & \text{for } i \in T' \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Then $\mathbb{1}_S$ is an eigenvector of M with eigenvalue 0. Furthermore, if (Y, α) is feasible for this B and satisfies condition (c) of Lemma 10, then the dual variable assignment (Y, B, α) is optimal, with objective value $4(\varepsilon_1 + \varepsilon_2)n$.

Thus, at this point, we know how to show that a feasible (Y, α) satisfying the conditions in Lemma 10 can be extended to get the dual variable B that ensures that the primal is integral. It now remains to show that given an assignment to Y that satisfies conditions in Lemma 10, and a B constructed thereof using Lemma 12, we can find an appropriate α so that (Y, B, α) is actually feasible.

The setting of B helps us exploit the following fact in showing that (Y, B, α) is feasible for some α.

▶ **Fact 13.** *If $M \in \mathbb{R}^{n \times n}$ is a symmetric matrix with eigenvector v having eigenvalue 0, then $M \succeq 0$ and $\operatorname{rank}(M) = n - 1 \iff \exists l > 0 \ : \ M \ + \ l \cdot vv^T \succ 0$.*

Thus, instead of showing that $M := (L + \alpha \mathbb{1}\mathbb{1}^T - B) \succeq 0$, we use the above fact with $v = \mathbb{1}_S$. By our setting for B, $\mathbb{1}_S$ is an eigenvector of M with eigenvalue 0. We show that there is an α that gives us that $M' := L + \alpha \left(\mathbb{1}\mathbb{1}^T + \mathbb{1}_S \mathbb{1}_S^T\right) - B \succ 0$ (by using condition (a) and (b) of Lemma 10).

This ensures M satisfies Lemma 11, completing the proof of Lemma 10. We leave the details to the full version of the paper.

2.2 Satisfying the sufficient condition

We are now left with the task of assigning appropriate weights Y_{ij} to the edges, so that Lemma 10's conditions are satisfied. Effectively, we only have to find weights within $G[S]$ and

$G[S']$, as Lemma 10 already fixes the rest. First, we recall that for any $X \in \mathbb{R}^n$, $X^T L(Y) X = \sum_{\{i,j\}\in E}(Y_{ij} + Y_{ji})(X_i - X_j)^2$. We drop the argument Y henceforth as it will be clear from context. The following observation is useful to keep in mind:

▶ **Observation 14.** *Given a (Y, B, α) dual solution, that satisfies all but the constraints $\sum_{j\in N(i)} Y_{ij} = 1$, having instead that $\forall i \in V \;:\; \sum_{j\in N(i)} Y_{ij} \leqslant 1$, we can produce a feasible solution (Y', B, α) of the same objective value.*

We will henceforth find values for the dual variable Y satisfying just the weaker constraint $\sum_{j\in N(i)} Y_{ij} \leqslant 1$. We first describe the following instructive case, in order to aid intuition.

▶ **Lemma 15.** *When $G[S]$ and $G[S']$ are complete graphs, there is a constant $\beta < 1$ and weights Y, such that when $\varepsilon \stackrel{\text{def}}{=} \max\{\varepsilon_1, \varepsilon_2\} \leqslant \beta$, the sufficient condition in Lemma 10 is satisfied, and hence the primal SDP is integral.*

Proof. For every pair i, j such that $i \in S \setminus T$ and $t \in T$, we set $Y_{it} = b := \frac{1}{\varepsilon n}$. The Y_{ij}'s within S' are set similarly. Let $Y_{ij} = 0$ for all other edges within $G[S]$, and $G[S']$. The constraint $\sum_{j\in N(i)} Y_{ij} \leqslant 1$ is satisfied as $b \times \Delta_T(i) \leqslant 1 \implies b\varepsilon n \leqslant 1$.

In order to prove integrality, we verify that (2) holds for the chosen value of b. Expanding out the term $X^T L X$ gives:

$$X^T L X \;\geqslant\; b \sum_{\substack{i\in S\setminus T\\ t\in T}} (X_i - X_t)^2 \;+\; b \sum_{\substack{i\in S'\setminus T'\\ t\in T'}} (X_i - X_t)^2 \,.$$

From the condition in Lemma 10, we get that the primal SDP is integral as long as $bn \geqslant 8c/(1-\varepsilon)$, which is true as long as $\frac{1-\varepsilon}{\varepsilon} > 8c$. This is true for ε being less than a small enough constant. ◀

Let us now consider the general case. We will focus on just S henceforth, as similar arguments will work for S' too, and the feasible solution can be constructed independently for either part. Observe that in contrast to the complete graph above, certain terms are missing in the expansion of $X^T L_{|S} X$: these terms are of the form

$$(X_i - X_t)^2 \qquad\qquad \forall i \in S \setminus T,\, t \in T : i \notin N(t)\,.$$

One way to recover these terms is to make use of the following observation:

▶ **Fact 16.** *For any $x_1, x_2, \ldots x_{l+1} \in \mathbb{R}$, we have $\sum_{i=1}^{l}(x_i - x_{i+1})^2 \geqslant \frac{1}{l}(x_1 - x_{l+1})^2$*

Flow Routing:

Fact 16 gives us a way to generate terms of the form $(X_i - X_t)^2$ using the edges present in the graph $G[S]$. In particular, we can generate a missing term of the form $(X_i - X_t)^2$, as a sum along a path $P = (i_1 = i, i_2, i_3, \ldots, i_l = t)$ in G of the terms $(X_{i_j} - X_{i_{j+1}})^2$, for every $j \in [l-1]$. Each of these terms occurs in the expansion of $X^T L X$. If we use an amount a of the weight of each edge on $\mathcal{P}$ in doing so, the final term has a coefficient of $\frac{a}{l}$, and this can be seen as i attempting to sending a 'flow' of magnitude a to t via $\mathcal{P}$.

Generating all the missing terms can now be formulated as a flow-routing problem using paths of length at most l (for some fixed l). The flows going from i to t generate the term $(X_i - X_t)^2$. Lemma 10 can therefore be restated as the problem of routing at least $c'l/n$ units of flow from every $i \in S \setminus T$ to $t \in T$. The constraint on the (directed) flow edges out of i is determined by the values Y_{ij}. The capacity of the edge $\{i, j\}$ in the direction $i \to j$ is

Y_{ij}, and the outdegree constraint states that every vertex can push out at most one unit of flow in total. Furthermore, a flow of 'a' units travelling along a path of distance l to reach t finally contributes only a/l, due to Fact 16. We state this idea formally below.

▶ **Lemma 17** (Flow routing problem). *Suppose we are given $G[S]$ and $G[S']$ with a feasible assignment Y for the edges. Consider a directed version of $G[S]$, where every edge $\{i, j\} \in E$ is replaced by the directed edges (i, j) and (j, i) with capacities Y_{ij} and Y_{ji} respectively. If for some $l \in \mathbb{N}$, and for every $i \in S \setminus T$ and $t \in T$, we can route a flow of $c'l/n$ from $i \to t$ using paths of length at most l in $G[S]$ (and similarly for $G[S']$), while obeying the (directed) capacity constraints on the edges, then we have:*

$$X^T LX \geqslant \frac{c'}{n}\left(\sum_{i\in S\setminus T}\sum_{t\in T}(X_i - X_t)^2 + \sum_{i\in S'\setminus T'}\sum_{t\in T'}(X_i - X_t)^2\right) \qquad \forall X \in \mathbb{R}^n$$

However, simply routing flows satisfying the above in an arbitrary $G[S]$ (or $G[S']$) turns out to be impossible. We carefully exploit the spectral expansion of $G[S \setminus T]$ (and $G[S' \setminus T']$), along with the randomness on the graph in $S \setminus T \times T$ while routing our flows to find our assignment for Y. Figure 2 summarizes briefly the dual values for Y that we finally use. We leave the details for the full version of the paper.

References

1 Emmanuel Abbe. Community detection and stochastic block models: recent developments. *arXiv preprint arXiv:1703.10146*, 2017.

2 Emmanuel Abbe, Afonso S Bandeira, Annina Bracher, and Amit Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *IEEE Transactions on Network Science and Engineering*, 1(1):10–22, 2014.

3 Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2016.

4 Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *IEEE 56th Annual Symp. on Foundations of Computer Science (FOCS), 2015*, pages 670–688. IEEE, 2015.

5 Emmanuel Abbe and Colin Sandon. Recovering communities in the general stochastic block model without knowing the parameters. In *Advances in neural information processing systems*, pages 676–684, 2015.

6 Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic bp, and the information-computation gap, 2017. arXiv: 1512.09080 To Appear in Communications on Pure and Applied Mathematics (2017).

7 Naman Agarwal, Afonso S Bandeira, Konstantinos Koiliaris, and Alexandra Kolla. Multisection in the stochastic block model using semidefinite programming. In *Compressed Sensing and its Applications*, pages 125–162. Springer, 2017.

8 Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. URL: `http://dx.doi.org/10.1007/BF02579166`, `doi:10.1007/BF02579166`.

9 Noga Alon and Vitali D Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

10 Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2), 2009. (Preliminary version in *36th STOC*, 2004). `doi:10.1145/1502793.1502794`.

11 Roberto Battiti and Marco Protasi. Approximate algorithms and heuristics for max-sat. In *Handbook of Combinatorial Optimization: Volume1–3*, pages 77–148, Boston, MA, 1999. Springer US. URL: `http://dx.doi.org/10.1007/978-1-4613-0303-9_2`, `doi:10.1007/978-1-4613-0303-9_2`.

12 Sergey Bobkov, Christian Houdré, and Prasad Tetali. λ_∞, Vertex Isoperimetry and Concentration. *Combinatorica*, 20(2):153–172, 2000.

13 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. URL: `http://dx.doi.org/10.1109/SFCS.1987.22`, `doi:10.1109/SFCS.1987.22`.

14 T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral Properties of Hypergraph Laplacian and Approximation Algorithms. *J. ACM*, 65(3):15:1–15:48, 2018. URL: `http://doi.acm.org/10.1145/3178123`, `doi:10.1145/3178123`.

15 Amin Coja-Oghlan. Colouring semirandom graphs. *Combinatorics, Probability and Computing*, 16(4):515–552, 2007. `doi:10.1017/S0963548306007917`.

16 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008. URL: `https://doi.org/10.1137/05064299X`, `arXiv:https://doi.org/10.1137/05064299X`, `doi:10.1137/05064299X`.

17 Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001.

18 Olivier Guédon and Roman Vershynin. Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165(3-4):1025–1049, 2016.

19 Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming: Extensions. *IEEE Transactions on Information Theory*, 62(10):5918–5937, 2016.

20 Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

21 Mark Jerrum and Gregory B Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(1):155–175, 1998.

22 George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. In *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, Supercomputing '95, New York, NY, USA, 1995. ACM. URL: `http://doi.acm.org/10.1145/224170.224229`, `doi:10.1145/224170.224229`.

23 George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December 1998. URL: `http://dx.doi.org/10.1137/S1064827595287997`, `doi:10.1137/S1064827595287997`.

24 Chiheon Kim, Afonso S. Bandeira, and Michel X. Goemans. Community detection in hypergraphs, spiked tensor models, and sum-of-squares. *arXiv preprint: arXiv:1705.02973 [cs.DS]*, 2017.

25 Victor Klee and George J Minty. How good is the simplex algorithm. In *Shisha, Oved. Inequalities III (Proc. of 3rd Symp. on Inequalities, UCLA)*, pages 159–175. Academic Press, California, 1972.

26 Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, November 1999. URL: `http://doi.acm.org/10.1145/331524.331526`, `doi:10.1145/331524.331526`.

27 Anand Louis and Yury Makarychev. Approximation Algorithms for Hypergraph Small Set Expansion and Small Set Vertex Expansion. In *Approximation, Randomization, and*

Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014), volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 339–355, 2014. `doi:10.4230/LIPIcs.APPROX-RANDOM.2014.339`.

28 Anand Louis and Prasad Raghavendra, 2014. Personal Communication.

29 Anand Louis, Prasad Raghavendra, and Santosh Vempala. The complexity of approximating vertex expansion. In *Proc. of the 54th Annual Symp. on Foundations of Computer Science*, FOCS '13, pages 360–369, Washington, DC, USA, 2013. IEEE Computer Society. URL: `http://dx.doi.org/10.1109/FOCS.2013.46`, `doi:10.1109/FOCS.2013.46`.

30 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proc. of the 44th Annual ACM Symp. on Theory of Computing*, STOC '12, pages 367–384. ACM, 2012. URL: `http://doi.acm.org/10.1145/2213977.2214013`, `doi:10.1145/2213977.2214013`.

31 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Constant factor approximation for balanced cut in the pie model. In *Proc. of the 46th Annual ACM Symp. on Theory of Computing*, STOC '14, pages 41–49, New York, NY, USA, 2014. ACM. URL: `http://doi.acm.org/10.1145/2591796.2591841`, `doi:10.1145/2591796.2591841`.

32 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Learning communities in the presence of errors. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1258–1291, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL: `http://proceedings.mlr.press/v49/makarychev16.html`.

33 Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proc. of the 46th Annual ACM Symp. on Theory of Computing*, STOC '14, pages 694–703, New York, NY, USA, 2014. ACM. `doi:10.1145/2591796.2591857`.

34 Frank D. McSherry. Spectral partitioning of random graphs. In *Proc. of the 42nd IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 529–537, Washington, DC, USA, 2001. IEEE Computer Society. URL: `http://dl.acm.org/citation.cfm?id=874063.875554`.

35 Ankur Moitra, William Perry, and Alexander S Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 828–841. ACM, 2016.

36 Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *Conference on Learning Theory*, pages 356–370, 2014.

37 Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for the planted bisection model. In *Proc. of the 47th Annual ACM Symp. on Theory of Computing*, STOC '15, pages 69–75, New York, NY, USA, 2015. ACM. URL: `http://doi.acm.org/10.1145/2746539.2746603`, `doi:10.1145/2746539.2746603`.

38 Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *Combinatorica*, 2017. URL: `https://doi.org/10.1007/s00493-016-3238-8`, `doi:10.1007/s00493-016-3238-8`.

39 Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 755–764, New York, NY, USA, 2010. ACM. URL: `http://doi.acm.org/10.1145/1806689.1806792`, `doi:10.1145/1806689.1806792`.