New Abilities and Limitations of Spectral Graph Bisection*†

Martin R. Schuster[1] **and Maciej Liśkiewicz**[2]

1 **Institute of Theoretical Computer Science, University of Lübeck, Lübeck, Germany**
2 **Institute of Theoretical Computer Science, University of Lübeck, Lübeck, Germany**

Abstract

Spectral based heuristics belong to well-known commonly used methods which determines provably minimal graph bisection or outputs "fail" when the optimality cannot be certified. In this paper we focus on Boppana's algorithm which belongs to one of the most prominent methods of this type. It is well known that the algorithm works well in the random *planted bisection model* – the standard class of graphs for analysis minimum bisection and relevant problems. In 2001 Feige and Kilian posed the question if Boppana's algorithm works well in the semirandom model by Blum and Spencer. In our paper we answer this question affirmatively. We show also that the algorithm achieves similar performance on graph classes which extend the semirandom model.

Since the behavior of Boppana's algorithm on the semirandom graphs remained unknown, Feige and Kilian proposed a new semidefinite programming (SDP) based approach and proved that it works on this model. The relationship between the performance of the SDP based algorithm and Boppana's approach was left as an open problem. In this paper we solve the problem in a complete way by proving that the bisection algorithm of Feige and Kilian provides exactly the same results as Boppana's algorithm. As a consequence we get that Boppana's algorithm achieves the optimal threshold for exact cluster recovery in the *stochastic block model*. On the other hand we prove some limitations of Boppana's approach: we show that if the density difference on the parameters of the planted bisection model is too small then the algorithm fails with high probability in the model.

1998 ACM Subject Classification I.1.2 Algorithms

Keywords and phrases Minimum Graph Bisection, Spectral Methods, Convex Programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2017.66

1 Introduction

The minimum graph bisection problem is one of the classical NP-hard problems [22]: for an undirected graph G the aim is to partition the set of vertices $V = \{1, \dots, n\}$ (n even) into two equal sized sets, such that the number of cut edges, i.e. edges with endpoints in different bisection sides, is minimized. The bisection width of a graph G, denoted by $\mathrm{bw}(G)$, is then the minimum number of cut edges in a bisection of G. Due to practical significance in VLSI design, image processing, computer vision and many other applications (see [30, 5, 46, 29, 31, 38]) and its theoretical importance, the problem has been the subject of a considerable amount of research from different perspectives: approximability [37, 4, 20, 19, 28], average-case

* A full version of the paper is available at `https://arxiv.org/abs/1701.01337`.
† This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) grant LI 634/4-1.

25th Annual European Symposium on Algorithms (ESA 2017).
Editors: Kirk Pruhs and Christian Sohler; Article No. 66; pp. 66:1–66:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

complexity [10], and parameterized algorithms [33, 44] including the seminal paper in this field by Cygan et al. [15] showing that the minimum bisection is fixed parameter tractable.

In this paper we consider polynomial-time algorithms that for an input graph either output the *provable* minimum-size bisection or "fail" when the optimality cannot be certified. The methods should work well for all (or almost all, depending on the model) graphs of particular classes, i.e. provide for them a certified optimum bisection, while for irregular, worst case instances the output can be "fail", what is justifiable. We investigate two well-studied graph models: the *planted bisection model* and its extension the *semirandom model* which are widely used to analyze and benchmark graph partitioning algorithms. We refer to [10, 16, 9, 6, 14, 18, 11, 34, 8, 12, 32] to cite some of the relevant works. Moreover, we consider the *regular graph model* introduced of Bui et al. [10] and a new extension of the semirandom model. For a (semi)random model we say that some property is satisfied with high probability (w.h.p.) if the probability that the property holds tends to 1 as the number of vertices $n \to \infty$.

In the planted bisection model, denoted as $\mathcal{G}_n(p,q)$ with parameters $1 > p = p(n) \geq q(n) = q > 0$, the vertex set $V = \{1, \dots, n\}$ is partitioned randomly into two equal sized sets V_1 and V_2, called the *planted bisection*. Then for every pair of vertices do independently: if both vertices belong to the same part of the bisection (either both belong to V_1 or both belong to V_2) then include an edge between them with probability p; If the two vertices belong to different parts, then connect the vertices by an edge with probability q. In the semirandom model for graph bisection [18], initially a graph G is chosen at random according to model $\mathcal{G}_n(p,q)$. Then a monotone adversary is allowed to modify G by applying an arbitrary sequence of the following monotone transformations: (1) The adversary may remove from the graph any edge crossing a minimum bisection; (2) The adversary may add to the graph any edge not crossing the bisection. Finally, in the regular random model, denoted as $\mathcal{R}_n(r,b)$, with $r = r(n) < n$ and $b = b(n) \leq (n/2)^2$, the probability distribution is uniform on the set of all graphs on V that are r-regular and have bisection width b.

The planted bisection model was first proposed in the sociology literature [27] under the name *stochastic block model* to study community detection problems in random graphs. In this setting, the planted bisection V_1, V_2 (as described above) models latent communities in a network and the goal here is to recover the communities from the observed graph. In the general case, the model allows some errors by recovering, multiple communities, and also that $p(n) < q(n)$. The community detection problem on the stochastic block model has been subject of a considerable amount of research in physics, statistics and computer science (see e.g. [1, 35] for current surveys). In particular, an intensive study has been carried out on providing lower bounds on $|p - q|$ to ensure recoverability of the planted bisection.

The main focus of our work is the bisection algorithm proposed by Boppana [9]. Though introduced almost three decades ago, the algorithm belongs still to one of the most important heuristics in this area. However, several basic questions concerning the algorithm's performance remain open. Using a spectral based approach, Boppana constructs an implementable algorithm which, assuming the density difference

$$p - q \geq c\sqrt{p \ln n}/\sqrt{n} \quad \text{for a certain constant } c > 0 \tag{1}$$

bisects $\mathcal{G}_n(p,q)$ optimally w.h.p. (certifying the optimality of the solutions). Remarkably, for a long time this was the largest subclass of graphs $\mathcal{G}_n(p,q)$ for which a minimum bisection could be found. Since under the assumption (1) the planted bisection is minimum w.h.p., Boppana's algorithm solves the recovery problem for the stochastic block model with two communities. Boppana's algorithm works well also on the regular graph model $\mathcal{R}_n(r,b)$, assuming that

$$r \geq 6 \quad \text{and} \quad b \leq o(n^{1-1/\lfloor (r/2+1)/2 \rfloor}). \tag{2}$$

In this paper we investigate the problem if, under assumption (1), Boppana's algorithm works well for the semirandom model. This question was posed by Feige and Kilian in [18] and remained open so far. In our work we answer the question affirmatively. We show also that Boppana's algorithm provides the same results as the algorithm proposed currently by Hajek, Wu, and Xu [25]. As a consequence we get that Boppana's algorithm achieves the optimal threshold for exact recovery in the stochastic block model with parameters $p = \alpha \log(n)/n$ and $q = \beta \log(n)/n$. On the other hand we show some limitations of the algorithm. One of the main results in this direction is that the density difference (1) is tight: we prove that if $p - q \leq o(\sqrt{p \cdot \ln n}/\sqrt{n})$ then the algorithm fails on $\mathcal{G}_n(p, q)$ w.h.p.

Our Results. The motivation of our research was to systematically explore graph properties which guarantee that Boppana's algorithm outputs a certified optimum bisection. Due to [9] we know that random graphs from $\mathcal{G}_n(p, q)$ and $\mathcal{R}_n(r, b)$ satisfy such properties w.h.p. under assumptions (1) and (2) on p, q, r, and b as discussed above. But, as we will see later, the algorithm works well also for instances which deviate significantly from such random graphs.

Our first technical contribution is a modification of the algorithm to cope with graphs of more than one optimum bisection, like e.g. hypercubes. The algorithm proposed originally by Boppana does not manage to handle such cases. Our modification is useful to work on wider classes of graphs.

In this paper we introduce a natural generalization of the semirandom model of Feige and Kilian [18]. Instead of $\mathcal{G}_n(p, q)$, we start with an arbitrary initial graph model $\mathcal{G}_n$, and then apply a sequence of the transformations by a monotone adversary as in [18]. We denote such a model by $\mathcal{A}(\mathcal{G}_n)$. One of our main positive results is that if Boppana's algorithm outputs the minimum-size bisection for graphs in $\mathcal{G}_n$ w.h.p., then the algorithm finds a minimum bisection w.h.p. for the adversarial graph model $\mathcal{A}(\mathcal{G}_n)$, too. As a corollary, we get that under assumption (1), Boppana's algorithm works well in the semirandom model, denoted here as $\mathcal{A}(\mathcal{G}_n(p, q))$, and, assuming (2), in $\mathcal{A}(\mathcal{R}_n(r, b))$ – the semirandom regular model. This solves the open problem posed by Feige and Kilian in [18]. To the best of our knowledge, Boppana's algorithm is the only method known so far, that finds (w.h.p.) provably optimum bisections on all of the above random graph classes.

Since the behavior of the algorithm on the (common) semirandom model $\mathcal{A}(\mathcal{G}_n(p, q))$ remained unknown so far, Feige and Kilian proposed in [18] a new semidefinite programming (SDP) based approach which works for semirandom graphs, assuming (1). The relationship between the performance of the SDP based algorithm and Boppana's approach was left in [18] as an open problem. Feige and Kilian conjecture that for every graph G, their objective function $h_p(G)$ to certify the bisection optimality and the lower bound computed in Boppana's algorithm give the same value. In our paper we answer this question affirmatively. To compare the algorithms, we provide a primal SDP formulation for Boppana's approach and prove that it is equivalent to the dual SDP of Feige and Kilian. Next we give a dual program to the primal formulation of Boppana's algorithm and prove that the optima of the primal and dual programs are equal to each other. Note that unlike linear programming, for semidefinite programs there may be a duality gap. Thus, we show that the bisection algorithm of Feige and Kilian provides exactly the same results as Boppana's algorithm. However, an important advantage of the spectral method by Boppana over the SDP based approach by Feige and Kilian is that the spectral method is practically implementable reducing the bisection problem for graphs with n vertices to computing minima of a convex function of n variables while the algorithm in [18] needs to solve a semidefinite program over n^2 variables.

From the result that the method by Feige and Kilian is equivalent to Boppana's we get, as a consequence, that Boppana's algorithm achieves the sharp threshold for exact cluster recovery in the stochastic block model which has been obtained recently by Abbe et al. [2] and independently by Mossel et al. [36]. In [2, 36] it is proved that in the (binary) stochastic block model, with $p = \alpha \log(n)/n$ and $q = \beta \log(n)/n$ for fixed constants $\alpha \neq \beta$, if $(\sqrt{\alpha} - \sqrt{\beta})^2 > 2$, the planted clusters can be exactly recovered (up to a permutation of cluster indices) with probability converging to one; if $(\sqrt{\alpha} - \sqrt{\beta})^2 < 2$, no algorithm can exactly recover the clusters with probability converging to one. Note, that the choice of p and q is well justified: Mossel et al. show that if $q < p = \log(n)/n$ then the exact recovery is impossible for these parameters. In [25] Hajek et al. proved that the SDP of Feige and Kilian achieves the optimal threshold, i.e. if $(\sqrt{\alpha} - \sqrt{\beta})^2 > 2$ the SDP reconstructs communities w.h.p. From our result we get, that Boppana's algorithm achieves the threshold, too.

To analyze limitations of the spectral approach we provide structural properties of the space of feasible solutions searched by the algorithm. This allows us to prove that if an optimal bisection contains some forbidden subgraphs, then Boppana's algorithm fails. Using these tools, we were able to show that if the density difference $p - q$ is asymptotically smaller than $\sqrt{p \cdot \ln n}/\sqrt{n}$ then Boppana's algorithm fails to determine a certified optimum bisection on $\mathcal{G}_n(p, q)$ w.h.p. Note that our impossibility result is not a direct consequence of the lower bound for the exact cluster recovery discussed above. For example, for $q = \mathcal{O}(1)/n$ and $p = \sqrt{\log n}/n$ from Mossel et al. [36] we know that for these parameters the exact recovery is impossible but obviously this does not imply that determining of a certified optimum bisection is impossible either.

Related Works. Spectral partitioning goes back to Fiedler [21], who first proposed to use eigenvectors to derive partitions. Spielman and Teng e.g. showed, that spectral partitioning works well on planar graphs [40, 41], although there are also graphs on which purely spectral algorithms perform poorly, as shown by Guattery and Miller [24].

Also other algorithms have been proven to work on the planted bisection model. Condon and Karp [14] developed a linear time algorithm for the more general l-partitioning problem. Their algorithm finds the optimal partition with probability $1 - \exp(-n^{\Theta(\varepsilon)})$ in the planted bisection model with parameters satisfying $p - q = \Omega(1/n^{1/2-\varepsilon})$. Carson and Impagliazzo [11] show that a hill-climbing algorithm is able to find the planted bisection w.h.p. for parameters $p - q = \Omega((\ln^3 n)/n^{1/4})$. Dyer and Frieze [16] provide a min-cut via degrees heuristic that, assuming $n(p - q) = \Omega(n)$ finds and certifies the minimum bisection w.h.p. Note, that the density difference (1) assumed by Boppana still outperforms the above ones. Moreover a disadvantage of the methods against Boppana's algorithm, except for the last one, is that they do not certify the optimality of the solutions. In [34] McSherry describes a spectral based heuristic that applied to $\mathcal{G}(p, q)$ finds a minimum bisection w.h.p if p and q satisfy assumption (1) but it does not certify the optimality. Importantly, the algorithms above, similarly as Boppana's method, solve the recovery problem for the stochastic block model with two communities.

In [12] Coja-Oghlan developed a new spectral-based algorithm which, on the planted partition model $\mathcal{G}_n(p, q)$, enables for a wider range of parameters than (1), certifying the optimality of its solutions. The algorithm [12] assumes that $p - q \geq \Omega(\sqrt{p \ln(np)}/\sqrt{n})$. If the parameters p and q describe non-sparse graphs, this condition is essentially the same as Boppana's assumption. For sparse graphs, however, Coja-Oghlan's constraint allows a larger subclass. For example, the algorithm works in $\mathcal{G}_n(p,q)$ for $q = \mathcal{O}(1)/n$ and $p = \sqrt{\log n}/n$. Due to results presented in our paper we know that Boppana's algorithm fails w.h.p. for such graphs. Interestingly, the condition on the density difference by Coja-Oghlan allows graphs for which the minimum bisection width is strictly smaller than the width of the planted bisection w.h.p. However, a drawback of Coja-Oghlan's algorithm is that to work well in the planted bisection model with *unknown* parameters p and q, the algorithm has to learn the parameters since it is based on the knowledge of values p and q. Also the performance of the algorithm on other families, like e.g. semirandom graphs and the regular random graphs $\mathcal{R}_n(r,b)$, is unknown. Recent research by Coja-Oghlan et al. [13] contributes to a better understanding of the planted bisection model and average case behavior of a minimum bisection.

The paper is organized as follows. The next section contains an overview over Boppana's algorithm. In Section 3 we define the adversarial graph model and show, that Boppana's algorithm works well on this class. In Section 4 we compare the algorithm to the SDP approach of Feige and Kilian. Next, in Section 5 we propose a modification of the algorithm to deal with non-unique optimum bisections. Finally, we develop a new analysis of the algorithm and use it to show some limitations of the method. We conclude the paper with a discussion. The proofs of most of the propositions presented in Sections 2 through 6 can be found in the full version [39].

2 Boppana's Graph Bisection Algorithm

In this section we fix definitions and notations used in our paper and we recall Boppana's algorithm and known facts on its performance. We need the details of the algorithm to describe its extension in the next section. For a given graph $G = (V, E)$, with $V = \{1, \dots, n\}$, Boppana defines a function f for all real vectors $x, d \in \mathbb{R}^n$ as

$$f(G,d,x) = \textstyle\sum_{\{i,j\}\in E} \frac{1-x_i x_j}{2} + \sum_{i\in V} d_i(x_i^2 - 1). \tag{3}$$

Call by $S \subset \mathbb{R}^n$ the subspace of all vectors $x \in \mathbb{R}^n$, with $\sum_i x_i = 0$. Based on f, the function g' is defined as follows

$$g'(G,d) = \min_{\|x\|^2 = n, x\in S} f(G,d,x), \tag{4}$$

where $\|x\|$ denotes the L_2 norm of x. Note that g' is invariant under shifting d, i.e. $g'(G, d + \beta(1,\dots,1)^T) = g'(G,d)$ for every $\beta \in \mathbb{R}$. Vector x is named a *bisection vector* if $x \in \{+1,-1\}^n$ and $\sum_i x_i = 0$. Such x determines a bisection of G of the cut width denoted as $\mathrm{cw}(x) = \sum_{\{i,j\}\in E} \frac{1-x_i x_j}{2}$. For a bisection vector x the function f takes the value (3) regardless of d. Minimization over all such x would give the minimum bisection width. Since g' uses a relaxated constraint we get $g'(G,d) \le \mathrm{bw}(G)$ where, recall, $\mathrm{bw}(G)$ denotes the bisection width of G. To improve the bound, Boppana tries to find some d which leads to a minimal decrease of the function value of g' compared to the bisection width:

$$h(G) = \max_{d\in\mathbb{R}^n} g'(G,d). \tag{5}$$

It is easy to see that for every graph G we have $h(G) \le \mathrm{bw}(G)$.

In order to compute g' efficiently, Boppana expresses the function in spectral terms. To describe this we need some definitions. Let I denote the n-dimensional identity matrix and let $P = I - \frac{1}{n}J$ be the projection matrix which projects a vector $x \in \mathbb{R}^n$ to the projection Px of vector x into the subspace S. Here, J denotes an $n \times n$ matrix of ones. For a matrix

$B \in \mathbb{R}^{n\times n}$, the matrix $B_S = PBP$ projects a vector $x \in \mathbb{R}^n$ to S, then applies B and projects the result again into S. Further, for $B \in \mathbb{R}^{n\times n}$ and $d \in \mathbb{R}^n$ we denote the sum of B's elements as $\text{sum}(B) = \sum_{ij} B_{ij}$ and by $\text{diag}(d)$ we denote the $n \times n$ diagonal matrix D with the entries of the vector d on the main diagonal, i. e. $D_{ii} = d_i$.

Now assume $B \in \mathbb{R}^{n\times n}$ is symmetric and let $B_S = PBP$. Denote by $\mathbb{R}^n_{\neq c\mathbf{1}}$ the real space $\mathbb{R}^n$ without the subspace spanned by the identity vector $\mathbf{1}$, i. e. $\mathbb{R}^n_{\neq c\mathbf{1}} = \mathbb{R}^n \setminus \{c\mathbf{1} : c \in \mathbb{R}\}$. We define $\lambda(B_S) = \max_{x\in\mathbb{R}^n_{\neq c\mathbf{1}}} \frac{x^T B_S x}{\|x\|^2}$. It is easy to see that if $\lambda(B_S) \geq 0$ then

$$\lambda(B_S) = \max_{x\in\mathbb{R}^n} \frac{x^T B_S x}{\|x\|^2} \tag{6}$$

i. e. $\lambda(B_S)$ is the largest eigenvalue of the matrix B_S. Vectors x that attain the maximum are exactly the eigenvectors corresponding to the largest eigenvalue $\lambda(B_S)$ of B_S.

Let G be an undirected graph with n vertices and adjacency matrix A. Let further $d \in \mathbb{R}^n$ be some vector and let $B = A + \text{diag}(d)$, then we define

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

In [9] it is shown that function g' can be expressed as $g'(G, d) = g(G, -4d)$. Since in the definition of h in (5) we maximize over all d, we can conclude that

$$h(G) = \max_{d\in\mathbb{R}^n} g(G, d) = \max_{d\in\mathbb{R}^n} \frac{\text{sum}(A + \text{diag}(d)) - n\lambda((A + \text{diag}(d))_S)}{4}. \tag{7}$$

Boppana's algorithm that finds and certifies an optimal bisection, works as follows:

Algorithm 1: Boppana's Algorithm

1. Compute $h(G)$: Numerically find a vector d^{opt} which maximizes $g(G, d)$. Let $D = \text{diag}(d^{\text{opt}})$. Use constraint $\sum_i d_i^{\text{opt}} = 2|E|$ to ensure $\lambda((A + D)_S) > 0$;
2. Construct a bisection: Let x be an eigenvector corresponding to the eigenvalue $\lambda((A + D)_S)$. Construct a bisection vector $\hat{x}$ by splitting at the median $\bar{x}$ of x, i.e. let $\hat{x}_i = +1$ if $x_i \geq \bar{x}$ and $\hat{x}_i = -1$ if $x_i < \bar{x}$. If $\sum_i \hat{x}_i > 0$, move (arbitrarily) $\frac{1}{2}\sum_i \hat{x}_i$ vertices i with $x_i = \bar{x}$ to part -1 letting $\hat{x}_i = -1$;
3. Output $\hat{x}$; If $\text{cw}(\hat{x}) = h(G)$ output "optimum bisection" else output "fail".

One can prove that g is concave and hence, the maximum in Step 1 can be found in polynomial time with arbitrary precision [23]. To analyze the algorithm's performance, Boppana proves the following, for a sufficiently large constant $c > 0$:

▶ **Theorem 1** (Boppana [9]). *Let G be a random graph from $\mathcal{G}_n(p, q)$, and let $p - q \geq c(\sqrt{p \ln n}/\sqrt{n})$. Then with probability $1 - \mathcal{O}(1/n)$, the bisection width of G equals $h(G)$.*

From this result one can conclude that the value $h(G)$ computed by the algorithm is, w.h.p., equal to the optimal bisection width of G. However, to guarantee that the algorithm works well one needs additionally to show that it also finds an optimal bisection:

▶ **Theorem 2.** *For random graphs G from $\mathcal{G}_n(p, q)$, with $p - q \geq c(\sqrt{p \ln n}/\sqrt{n})$, Boppana's algorithm certifies the optimality of $h(G)$ revealing w.h.p. bisection vector $\hat{x}$ of $\text{cw}(\hat{x}) = h(G)$.*

To prove this theorem one first has to revise carefully the proof of Theorem 1 in [9] and show that w.h.p. the multiplicity of the largest eigenvalue of the matrix $(A + D)_S$ in Step 1 is 1. This was observed already in [7]. Next we need the following property:

▶ **Lemma 3.** *Let G be a graph with $h(G) = \mathrm{bw}(G)$ and let $d^{\mathrm{opt}} \in \mathbb{R}^n$ s. t. $g(G, d^{\mathrm{opt}}) = \mathrm{bw}(G)$ and $\sum_i d_i^{\mathrm{opt}} \geq 4\,\mathrm{bw}(G) - 2|E|$. Denote further by $B^{\mathrm{opt}} = A + \mathrm{diag}(d^{\mathrm{opt}})$. Then every optimum bisection vector y is an eigenvector of B_S^{opt} corresponding to the largest eigenvalue $\lambda(B_S^{\mathrm{opt}})$.*

(The proof of Lemma 3, as the proofs of most of the remaining propositions presented in this paper, can be found in the full version [39].) This completes the proof that the algorithm works well on random graphs from $\mathcal{G}_n(p, q)$.

3 Bisections in Adversarial Models

We introduce the *adversarial model*, denoted by $\mathcal{A}(\mathcal{G}_n)$, as a generalization of the semirandom model in the following way. Let $\mathcal{G}_n$ be a graph model, i.e. a class of graphs with distributions over graphs of n nodes (n even). In the model $\mathcal{A}(\mathcal{G}_n)$, initially a graph G is chosen at random according to $\mathcal{G}_n$. Let (Y_1, Y_2) be a fixed, but arbitrary optimal bisection of G. Then, similarly as in [18], a monotone adversary is allowed to modify G by applying an arbitrary sequence of the following monotone transformations: The adversary may

1. remove from the graph any edge $\{u, v\}$ crossing the bisection ($u \in Y_1$ and $v \in Y_2$);
2. add to the graph any edge $\{u, v\}$ not crossing the bisection ($u, v \in Y_1$ or $u, v \in Y_2$).

For example, $\mathcal{A}(\mathcal{G}_n(p, q))$ is the semirandom model as defined in [18].

We will prove that Boppana's algorithm works well for graphs from adversarial model $\mathcal{A}(\mathcal{G}_n)$ if the algorithm works well for $\mathcal{G}_n$. First we show that, if the algorithm is able to find an optimal bisection size of a graph, we can add edges within the same part of an optimum bisection and that we can remove cut edges, and the algorithm will still work. This solves the open question of Feige and Kilian [18].

Note that the result follows alternatively from Corollary 11 (presented in Section 4) that the SDPs of [18] are equivalent to Boppana's optimization function and form the property proved in [18] that the objective function of the dual SDP of Feige and Kilian preserves minimal bisection regardless of monotone transformations. The aim of this section is to give a direct proof of this property for Boppana's algorithm.

▶ **Theorem 4.** *Let $G = (V, E)$ be a graph with $h(G) = \mathrm{bw}(G)$. Consider some optimum bisection Y_1, Y_2 of G.*

1. *Let u and v be two vertices within the same part, i.e. $u, v \in Y_1$ or $u, v \in Y_2$, and let $G' = (V, E \cup \{\{u, v\}\})$. Then $h(G') = \mathrm{bw}(G')$.*
2. *Let u and v be two vertices in different parts, i.e. $u \in Y_1$ and $v \in Y_2$, with $\{\{u, v\}\} \in E$ and let $G' = (V, E \setminus \{\{u, v\}\})$. Then $h(G') = \mathrm{bw}(G) - 1 = \mathrm{bw}(G')$.*

Sketch of Proof. In order to prove the first part of the theorem, i.e. when we add an edge $\{u, v\}$, let A and A' denote the adjacency matrices of G and G', respectively. It holds $A' = A + A^{\Delta}$ with $A^{\Delta}_{uv} = A^{\Delta}_{vu} = 1$ and zero everywhere else. The main idea is now, that we can derive a new optimal correction vector d' for G' based on the optimal correction vector d^{opt} for G. We set $d' = d^{\mathrm{opt}} + d^{\Delta}$ with $d_i^{\Delta} = \begin{cases} -1 & \text{if } i = u \text{ or } i = v, \\ 0 & \text{else.} \end{cases}$

The known changes in the adjacency matrix as well as the derived correction vector allow us to compute $g(G', d')$ and to show that $g(G', d') = \mathrm{bw}(G')$. The proof of the second part of the theorem works analogously. The complete proof can be found in the full version [39]. ◀

▶ **Theorem 5.** *If Boppana's algorithm finds a minimum bisection for a graph model $\mathcal{G}_n$ w.h.p., then it finds a minimum bisection w.h.p. for the adversarial model $\mathcal{A}(\mathcal{G}_n)$, too.*

As a direct consequence, we obtain the following corollary regarding the semirandom graph model considered by Feige and Kilian:

▶ **Corollary 6.** *Under assumption* (1) *on* p *and* q*, Boppana's algorithm computes the minimum bisection in* $\mathcal{A}(\mathcal{G}_n(p,q))$*, i.e. in the semirandom model, w.h.p.*

In [9], Boppana also considers random regular graphs $\mathcal{R}_n(r,b)$, where a graph is chosen uniformly over the set of all r-regular graphs with bisection width b. He shows that his algorithm works w.h.p. on this graph under the assumption that $b = o(n^{1-1/\lfloor (r+1)/2 \rfloor})$. We can now define the semirandom regular graph model as adversarial model $\mathcal{A}(\mathcal{R}_n(r,b))$. Applying Theorem 5, we obtain

▶ **Corollary 7.** *Under assumption* (1) *on* p *and* q*, Boppana's algorithm computes the minimum bisection in the semirandom regular model w.h.p.*

4 SDP Characterizations of the Graph Bisection Problem

Feige and Kilian express the minimum-size bisection problem for an instance graph G as a semidefinite programming problem (SDP) with solution $h_p(G)$ and prove that the function $h_d(G)$, which is the solution to the dual SDP, reaches $\text{bw}(G)$ w.h.p. Since $\text{bw}(G) \geq h_p(G) \geq h_d(G)$, they conclude that $h_p(G)$ as well reaches $\text{bw}(G)$ w.h.p. The proposed algorithm computes $h_p(G)$ and reconstructs the minimum bisection of G from the optimum solution of the primal SDP. The authors conjecture in [18, Sec. 4.1.] the following: "Possibly, for every graph G, the function $h_p(G)$ and the lower bound $h(G)$ computed in Boppana's algorithm give the same value, making the lemma that $h_p(G) = \text{bw}(G)$ w.h.p. a restatement of the main theorem of [9]. In this section we answer this question affirmatively.

The semidefinite programming approach for optimization problems was studied by Alizadeh [3], who as first provided an equivalent SDP formulation of Boppana's algorithm. Before we give an SDP introduced by Feige an Kilian, we recall briefly some basic definitions and provide an SDP formulation for Boppana's approach. On the space $\mathbb{R}^{n\times m}$ of $n\times m$ matrices, we denote by $A\bullet B$ an inner product of A and B defined as $A\bullet B = \text{tr}(AB) = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ij}$, where $\text{tr}(C)$ is the trace of the (square) matrix C. Let A be an $n \times n$ symmetric real matrix, then A is called symmetric positive semidefinite (SPSD) if A is symmetric, i.e. $A^T = A$, and for all real vectors $v \in \mathbb{R}^n$ we have $v^T A v \geq 0$. This property is denoted by $A \succeq 0$. Note that the eigenvalues of a symmetric matrix are real.

For given real vector $c \in \mathbb{R}^n$ and $m+1$ symmetric matrices $F_0,\ldots,F_m \in \mathbb{R}^{n\times n}$ an SDP over variables $x \in \mathbb{R}^n$ is defined as

$$\min_x c^T x \quad \text{subject to} \quad F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0. \tag{8}$$

The dual program associated with the SDP (for details see e.g. [45]) is the program over the variable matrix $Y = Y^T \in \mathbb{R}^{n\times n}$:

$$\max_Y \; -F_0 \bullet Y \quad \text{subject to} \quad \forall i: \; F_i \bullet Y = c_i \quad \text{and} \quad Y \succeq 0. \tag{9}$$

It is known that the optimal value of the maximization dual SDP is never larger than the optimal value of the minimization primal counterpart. However, unlike linear programming, for semidefinite programs there may be a duality gap, i.e. the primal and/or dual might not attain their respective optima.

To prove that for any graph G Boppana's function $h(G)$ gives the same value as $h_p(G)$ we formulate the function h as a (primal) SDP. We provide also its dual program and prove that the optimum solutions of primal and dual are equal in this case. Then we show that the dual formulation of the Boppana's optimization is equivalent to the primal SDP defined by Feige and Kilian [18].

Below, $G = (V, E)$ denotes a graph, A the adjacency matrix of G and for a given vector d, as usually, let $D = \mathrm{diag}(d)$, for short. We provide the SDP for the function h (Eq. (7)) that differ slightly from that one given in [3].

▶ **Proposition 8.** *For any graph $G = (V, E)$, the objective function*

$$h(G) \;=\; \max_{d\in\mathbb{R}^n} \frac{\mathrm{sum}(A + D) - n\lambda((A + D)_S)}{4}$$

maximized by Boppana's algorithm can be characterized as an SDP as follows:

$$\begin{cases} p(G) = \min\limits_{z\in\mathbb{R}, d\in\mathbb{R}^n} (nz - \mathbf{1}^T d) \quad \textit{subject to} \\ zI - A + \frac{JA+AJ}{n} - \frac{\mathrm{sum}(A)J}{n^2} - D + \frac{\mathbf{1}d^T + d\mathbf{1}^T}{n} - \frac{\mathrm{sum}(D)J}{n^2} \;\succeq\; 0, \end{cases} \tag{10}$$

with the relationship $h(G) = \frac{|E|}{2} - \frac{1}{4}p(G)$. The dual program to the program (10) *can be expressed as follows:*

$$\begin{cases} d(G) = \max\limits_{Y\in\mathbb{R}^{n\times n}} \left(A \bullet Y - \frac{1}{n}\sum_j \deg(j) \sum_i y_{ij} - \frac{1}{n}\sum_i \deg(i) \sum_j y_{ij} + \frac{1}{n^2}\sum_{i,j} y_{ij}\right) \\ \textit{subject to} \\ \qquad\qquad \sum_i y_{ii} \;=\; n, \\ \forall i \quad y_{ii} - \frac{1}{n}\sum_j y_{ji} - \frac{1}{n}\sum_j y_{ij} + \frac{1}{n^2}\sum_{k,j} y_{kj} \;=\; 1, \\ \qquad\qquad Y \;\succeq\; 0. \end{cases} \tag{11}$$

Using these formulations we prove that the primal and dual SDPs attain the same optima.

▶ **Theorem 9.** *For the semidefinite programs of Proposition 8 the optimal value p^* of the primal SDP* (10) *is equal to the optimal value d^* of the dual SDP* (11)*. Moreover, there exists a feasible solution (z, d) achieving the optimal value p^*.*

Proof. Consider the primal SDP (10) of Boppana in the form

$$\min_{z\in\mathbb{R}, d\in\mathbb{R}^n} z \quad \text{s.t.} \quad zI - M(d) \;\succeq\; 0,$$

with $M(d) = P(A + \mathrm{diag}(d))P - \frac{\mathbf{1}^T d}{n} I$ and, recall, $P = I - \frac{J}{n}$. Note that this formulation is equivalent to (10), as we have shown in the proof of Proposition 8. We show that this primal SDP problem is strictly feasible, i.e. that there exists an z' and an d' with $z'I - M(d') \succ 0$. To this aim we choose an arbitrary d' and then some $z' > \lambda(M(d'))$. From [45, Thm. 3.1], it follows that the optima of primal and dual obtain the same value.

To prove the second part of the theorem, i.e. there exists a feasible solution achieving the optimal value p^*, consider the following. The function $h(G)$ maximizes $g(G, d)$ over vectors $d \in \mathbb{R}^n$, while d can be restricted to vectors of mean zero. The function g is convex and goes to $-\infty$ for vectors d with some component going to ∞. Thus, g reaches its maximum at some finite d^{opt}. Now we choose $d = d^{\mathrm{opt}}$ and $z = \lambda(M(d^{\mathrm{opt}}))$. Clearly, this solution is feasible and obtains the optimal value p^*. ◀

For a graph $G = (V, E)$, Feige and Kilian express the minimum bisection problem as an SDP over an $n \times n$ matrix Y as follows:

$$h_p(G) = \min_{Y \in \mathbb{R}^{n \times n}} h_Y(G) \quad \text{s.t.} \quad \forall i \; y_{ii} = 1, \; \sum_{i,j} y_{ij} = 0, \text{ and } Y \succeq 0, \tag{12}$$

where $h_Y(G) = \sum_{\substack{\{i,j\} \in E \\ i<j}} \frac{1-y_{ij}}{2}$. For proving that the SDP takes as optimum the bisection width w.h.p. on $\mathcal{G}_n(p, q)$, the authors consider the dual of their SDP:

$$h_d(G) = \max_{x \in \mathbb{R}^n} \left(\frac{|E|}{2} + \frac{1}{4} \sum_i x_i \right) \quad \text{s.t.} \quad M = -A - x_0 J - \text{diag}(x) \succeq 0, \tag{13}$$

where A is the adjacency matrix of G. They show that the dual takes the value of the bisection width w.h.p. and bounds the optimum of the primal SDP. Although we know that their SDP and Boppana's algorithm both work well on $\mathcal{G}_n(p, q)$, it was open so far how they are related to each other. Below we answer this question showing that the formulations are equivalent. We start with the following:

▶ **Theorem 10.** *The primal SDP* (12) *is equivalent to the dual SDP* (11)*, with the relationship* $h_p(G) = \frac{|E|}{2} - \frac{1}{4} d(G)$*.*

From Theorems 9 and 10 we get

▶ **Corollary 11.** *Let G be an arbitrary graph. Then for the lower bound $h(G)$ of Boppana's algorithm and for the objective functions $h_p(G)$ of the primal SDP* (12)*, resp. $h_d(G)$ of the dual SDP* (12) *of Feige and Kilian [18] it is true*

$$h(G) = h_p(G) = h_d(G).$$

Thus, the both algorithms provide for any graph G the same objective value. We want to point out another important fact: the bisection algorithm proposed in [18] use an SDP formulation, where the variables are a matrix with dimension $n \times n$. Thus, there are n^2 variables for a graph with n vertices. In contrast, Boppana's algorithm uses n variables in the convex optimization problem. If we consider the dual SDP, we again have only $n + 1$ variables. However, due to Corollary 11, we can't be better than Boppana's algorithm.

Abbe et al. [2] and independently Mossel et al. [36] have shown, that there is a sharp threshold phenomenon when considering the $\mathcal{G}_n(p, q)$ model with $p = \alpha \log(n)/n$ and $q = \beta \log(n)/n$ for fixed constants α, β, $\alpha > \beta$. Exact recovery of the planted bisection is possible if and only if $(\sqrt{\alpha} - \sqrt{\beta})^2 > 2$ (see e.g. [36] for a formal definition of exact cluster recovery problem). Hajek et al. [25] show, than an SDP equivalent to the one of Feige and Kilian achieves this bound. Since, due to Corollary 11, we know that the SDP is equivalent to Boppana's algorithm, we conclude that also Boppana's algorithm achieves the optimal threshold for finding and certifying the optimal bisection in the considered model. We get:

▶ **Theorem 12.** *Let α and β, $\alpha > \beta$, be constants. Consider the graph model $\mathcal{G}_n(p, q)$ with $p = \alpha \log(n)/n$ and $q = \beta \log(n)/n$. Then, as $n \to \infty$, if $(\sqrt{\alpha} - \sqrt{\beta})^2 > 2$, Boppana's algorithm recovers the planted bisection w.h.p. If $(\sqrt{\alpha} - \sqrt{\beta})^2 < 2$, no algorithm is able to recover the planted bisection w.h.p.*

Proof. The second part of the theorem is exactly the statement from [2]. The first part, i.e. that Boppana's algorithm is able to recover the bisection, follows from [25, Thm. 2]. Hajek et al. show, that for $(\sqrt{\alpha} - \sqrt{\beta})^2 > 2$ the SDP of Feige and Kilian obtain the optimal solution. Due to Theorem 10, the same holds for Boppana's algorithm. ◀

5 Certifying Non-Unique Optimum Bisections

From Section 2 we know that if the bound $h(G)$ is tight and the bisection of minimum size is unique, or more precisely the multiplicity of the largest eigenvector of B_S is 1, Boppana's algorithm is able to certify the optimality of the resulting bisection. We say that a graph G has a unique optimum bisection if there exists a unique, up to the sign, bisection vector x such that $\mathrm{cw}(x) = \mathrm{cw}(-x) = \mathrm{bw}(G)$. In this paper we also investigate families of graphs, different than random graphs $\mathcal{G}_n(p, q)$, for which Boppana's approach works well. To this aim we show a modification which handles cases such that $h(G) = \mathrm{bw}(G)$ but for which no unique bisection of minimum size exists. As we will see later hypercubes satisfy these two conditions. We present our algorithm below. Note that if the multiplicity of the largest eigenvalue of B_S^{opt} is 1, then the algorithm outputs the same result as in the original algorithm by Boppana.

1. Perform Step 1 of Algorithm 1; Let x be an eigenvector corresponding to the eigenvalue $\lambda((A + D)_S)$ and let k be the multiplicity of the largest eigenvalue of $(A + D)_S$
2. If $k = 1$ then construct a bisection vector $\hat{x}$ by splitting at the median $\bar{x}$ as in Step 2 of Algorithm 1; Next output $\hat{x}$ and if $\mathrm{cw}(\hat{x}) = h(G)$ output "optimum bisection" else output "fail"; If $k > 1$ then perform the steps below
3. Let $M \in \mathbb{R}^{n\times k}$ be the matrix with k linear independent eigenvectors corresponding to this largest eigenvalue; Transform the matrix to the reduced column echelon form, i. e. there are k rows which form an identity matrix, s.t. M still spans the same subspace
4. Brute force: for every combination of k coefficients from $\{+1, -1\}$ take the linear combination of the k vectors of M with the coefficients and verify if the resulting vector x is a bisection vector, i.e. $x \in \{+1, -1\}^n$ with $\sum_i x_i = 0$. If yes and if $\mathrm{cw}(x) = h(G)$ then output x and continue. This needs 2^k iterations
5. If in Step 4 no bisection vector x is given then output "fail".

▶ **Theorem 13.** *If $h(G) = \mathrm{bw}(G)$ then the algorithm above reconstructs all optimal bisections. Every achieved bisection vector corresponds to an optimal bisection.*

The eigenvalues for the family of hypercubes are explicitly known [26]. Hence, we can verify that the bound $h(G)$ is tight and Boppana's algorithm with the modification above works, i.e. finds an optimal bisection. For a hypercube H_n with n vertices we have $h(H_n) = g(H_n, (2 - \log n)\mathbf{1}) = n/2 = \mathrm{bw}(H_n)$. Since the hypercube with n vertices has $\log n$ optimal bisections and the largest eigenspace of B_S has multiplicity $\log n$, the brute force part in our modification of Boppana's algorithm results in a linear factor of n for the overall runtime. Thus, the algorithm runs in polynomial time. With the results from Section 3 we can extend this result and obtain, that Boppana's algorithm with our modification works on adversarially modifieded hypercubes as well.

6 The Limitations of the Algorithm

Boppana shows, that his algorithm works well on some classes of random graphs. However, we do not know which graph properties force the algorithm to fail. For example, for the considered planted bisection model, we require a small bisection width. On the other hand, as we have seen in Section 5 Boppana's algorithm works for the hypercubes and their semirandom modifications – graphs that have large minimum bisection sizes.

In the following, we present newly discovered structural properties from inside the algorithm, which provide a framework for a better analysis of the algorithm itself. Let y be a bisection vector of G. We define

$$d^{(y)} = -\,\mathrm{diag}(y)Ay. \tag{14}$$

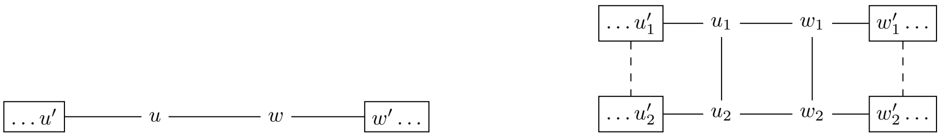

Figure 1 Forbidden graph structures as in Corollary 17 (left) and in Corollary 18 (right).

An equivalent but more intuitive characterization of $d^{(y)}$ is the following: $d_i^{(y)}$ is the difference between the number of adjacent vertices in other partition as vertex i and the number of adjacent vertices in same partition as i.

▶ **Lemma 14.** *Let G be a graph with $h(G) = \mathrm{bw}(G)$ and assume there is more than one optimum bisection in G. Then (up to constant translation vectors $c\mathbf{1}$) there exists a unique vector d^{opt} with $g(G, d^{\mathrm{opt}}) = \mathrm{bw}(G)$. Additionally, for every bisection vector y of an arbitrary optimum bisection in G there exists a unique $\alpha^{(y)}$ and the corresponding $d^{(y)}$, with $g(G, d^{(y)} + \alpha^{(y)} y) = \mathrm{bw}(G)$.*

Thus, if there are two optimum bisections represented by y and y' with $d^{(y)} \neq d^{(y')}$, then the difference of the d-vectors in component i is only dependent on y_i and y'_i, since we have $d^{(y)} - d^{(y')} = \beta' y' - \beta y$ for some constants β and β'. This structural property allows us to show the following limitation for the sparse planted partition model $\mathcal{G}_n(p, q)$.

▶ **Theorem 15.** *The algorithm of Boppana fails w.h.p. in the subcritical phase from [12], defined as $n(p-q) = \sqrt{np \cdot \gamma \ln n}$, for real $\gamma > 0$.*

In the planted partition model $\mathcal{G}_n(p, q)$, if the graphs are dense, e.g. $p = 1/n^c$ for a constant c with $0 < c < 1$, the constraints for the density difference $p - q$ assumed in Boppana's [9] and Coja-Oghlan's [12] algorithms are essentially the same. However for sparse graphs, e.g. such that $q = \mathcal{O}(1)/n$, the situation changes drastically. Now, e.g. $p = \sqrt{\log n}/n$ satisfy Coja-Oghlan's constraint $p - q \geq \Omega(\sqrt{p \ln(pn)}/\sqrt{n})$ but the condition on the difference $p - q$ assumed by Boppana is not true any more. Theorem 15 shows that Boppana's algorithm indeed fails under this setting. The proof of this theorem relies on the following observation, which can be derived from our newly discovered structural properties from above.

▶ **Lemma 16.** *Let G be a graph with $h(G) = \mathrm{bw}(G)$ and let (Y_1, Y_{-1}) be an arbitrary optimal bisection. Then, for each pair of vertices $v_i \in Y_i$, $i \in \{1, -1\}$, not connected by an edge ($\{v_i, v_{-i}\} \notin E$), we have: If $e(v_i, Y_i) = e(v_i, Y_{-i})$ for $i \in \{1, -1\}$ (the vertices have balanced degree), then $N(v_i) = N(v_{-i})$, i.e. both vertices have the same neighbors.*

I.e. if we have two balanced vertices in different parts of an optimal bisection, not connected by an edge, then the two vertices must have the same neighborhood as a necessary criterion for Boppana's algorithm to work. In the subcritical phase in Theorem 15, there exist most likely many of such pairs of vertices, but they are unlikely to have all even the same degree.

We can also provide forbidden substructures, which make Boppana's algorithm fail. This is e.g. the case, when the graph contains a path segment located on an optimal bisection:

▶ **Corollary 17.** *Let G be a graph, as illustrated in Fig. 1 (left), with $n \geq 10$ vertices containing a path segment $\{u', u\}, \{u, w\}, \{w, w'\}$, where u and w have no further edges. If there is an optimal bisection y, s. t. $y_u = y_{u'} = +1$ and $y_w = y_{w'} = -1$ (i. e. $\{u, w\}$ is a cut edge), then $h(G) < \mathrm{bw}(G)$.*

To prove this corollary, we use the more general but more technical Lemma 22 (Appendix of the full version [39]) with parameters $\tilde{C}_{+1} = \{u\}$ and $\tilde{C}_{-1} = \{w\}$. The result can also be applied for $2 \times c$ lattices:

▶ **Corollary 18.** *Let G be a graph with $n \geq 10c$ vertices containing a $2 \times c$ lattice with vertices u_i and w_i, as illustrated in Fig. 1 (right). (The construction is similar to the corollary above, but now we have a lattice instead of a single cut edge.) If there is an optimal bisection y, s. t. $y_{u_i} = y_{u'_i} = +1$ and $y_{w_i} = y_{w'_i} = -1$, then $h(G) < \mathrm{bw}(G)$.*

The algorithm fails if there are isolated vertices in both parts of an optimal bisection:

▶ **Theorem 19.** *Let G be a graph with $h(G) = \mathrm{bw}(G)$. Let G' be the graph G with two additional isolated vertices, then $h(G') \leq h(G) - \frac{4\,\mathrm{bw}(G)}{n^2}$.*

7 Discussion and Open Problems

Boppana's spectral method is a practically implementable heuristic. Computing eigenvalues and eigenvectors is well-studied and can be done very efficiently. Falkner, Rendl and Wolkowicz [17] show in a numerical study that using spectral techniques for graph partitioning is very robust and upper and lower bounds for the bisection width can be obtained such that the relative gap is often just a few percentage points apart. In [43] and [42], Tu, Shieh and Cheng present numerical experiments including results for Boppana's algorithm. They verify that the algorithm indeed has good average case behavior over certain probability distributions on graphs. We conducted further experiments on the graph model $\mathcal{R}_n(r, b)$ which indicated, that Boppana's algorithm also works for $r = 5$, but not for $r = 3$ and $r = 4$. An interesting question arising is, which properties of 3- and 4-regular graphs from the planted bisection model let the algorithm fail.

References

1. Emmanuel Abbe. Community detection and stochastic block models: recent developments. *arXiv preprint arXiv:1703.10146*, 2017.
2. 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.
3. Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.
4. Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proc. of the 27th Annual ACM Symposium on Theory of Computing (STOC)*, pages 284–293. ACM, 1995.
5. Sandeep N Bhatt and Frank Thomson Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, 1984.
6. Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19(2):204–234, 1995.
7. Robert D. Blumofe. Spectral methods for bisecting graphs. Unpublished Manuscript, 1993.
8. Béla Bollobás and Alex D. Scott. Max cut for random graphs with a planted partition. *Combinatorics, Probability and Computing*, 13(4-5):451–474, 2004.
9. Ravi B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proc. of the 28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 280–285. IEEE Computer Society, 1987. `doi:10.1109/SFCS.1987.22`.
10. T. N. Bui, S. Chaudhuri, F. T. Leighton, and M. Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2):171–191, 1987. `doi:10.1007/BF02579448`.
11. Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 903–909. SIAM, 2001.

12 Amin Coja-Oghlan. A spectral heuristic for bisecting random graphs. In *Proc. of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 850–859. SIAM, 2005. URL: `http://dl.acm.org/citation.cfm?id=1070432.1070552`.

13 Amin Coja-Oghlan, Charilaos Efthymiou, and Nor Jaafari. Local convergence of random graph colorings. In *Proc. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (APPROX/RANDOM)*, volume 40 of *LIPIcs*, pages 726–737, 2015. URL: `http://arxiv.org/abs/1505.02985`.

14 Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

15 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, MichałPilipczuk, and Saket Saurabh. Minimum bisection is fixed parameter tractable. In *Proc. of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 323–332. ACM, 2014. `doi:10.1145/2591796.2591852`.

16 Martin E. Dyer and Alan M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, 1989.

17 Julie Falkner, Franz Rendl, and Henry Wolkowicz. A computational study of graph partitioning. *Mathematical Programming*, 66(1-3):211–239, 1994. `doi:10.1007/BF01581147`.

18 Uriel Feige and Joe Kilian. Heuristics for semirandom graph problems. *Journal of Computer and System Sciences*, 63(4):639–671, 2001. `doi:10.1006/jcss.2001.1773`.

19 Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM Journal on Computing*, 31(4):1090–1118, April 2002. `doi:10.1137/S0097539701387660`.

20 Uriel Feige, Robert Krauthgamer, and Kobbi Nissim. Approximating the minimum bisection size. In *Proc. of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 530–536. ACM, 2000.

21 Miroslav Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

22 Michael R. Garey, David S. Johnson, and Larry Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

23 M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. `doi:10.1007/BF02579273`.

24 Stephen Guattery and Gary L. Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19(3):701–719, July 1998. `doi:10.1137/S0895479896312262`.

25 Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

26 Frank Harary, John P. Hayes, and Horng-Jyh Wu. A survey of the theory of hypercube graphs. *Computers and Mathematics with Applications*, 15(4):277–289, 1988.

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

28 Subhash Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

29 Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (ToG)*, 22(3):277–286, 2003.

30 Thomas Lengauer. *Combinatorial algorithms for integrated circuit layout*. Springer Science & Business Media, 2012.

31 Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

32 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proc. of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 367–384. ACM, 2012.

33 Dániel Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.

34 Frank McSherry. Spectral partitioning of random graphs. In *Proc. of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 529–537. IEEE, 2001.

35 Cristopher Moore. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. *arXiv preprint arXiv:1702.00467*, 2017.

36 Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for the planted bisection model. In *Proc. of the 47th ACM Symposium on Theory of Computing (STOC)*, pages 69–75. ACM, 2015.

37 Huzur Saran and Vijay V. Vazirani. Finding k cuts within twice the optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995.

38 Kirk Schloegel, George Karypis, and Vipin Kumar. *Graph partitioning for high performance scientific simulations*. Army High Performance Computing Research Center, 2000.

39 Martin R. Schuster and Maciej Liśkiewicz. New abilities and limitations of spectral graph bisection. *CoRR*, 2017. URL: `https://arxiv.org/abs/1701.01337`.

40 Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proc. of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 96–105. IEEE Computer Society, 1996. URL: `http://dl.acm.org/citation.cfm?id=874062.875505`.

41 Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2):284–305, 2007. `doi:10.1016/j.laa.2006.07.020`.

42 Chih-Chien Tu and Hsuanjen Cheng. Spectral methods for graph bisection problems. *Computers & operations research*, 25(7):519–530, 1998. `doi:10.1016/S0305-0548(98)00021-5`.

43 Chih-Chien Tu, Ce-Kuen Shieh, and Hsuanjen Cheng. Algorithms for graph partitioning problems by means of eigenspace relaxations. *European Journal of Operational Research*, 123(1):86–104, 2000. `doi:10.1016/S0377-2217(99)00060-0`.

44 René van Bevern, Andreas Emil Feldmann, Manuel Sorge, and Ondřej Suchý. On the parameterized complexity of computing graph bisections. In *Proc. International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 76–87. Springer, 2013.

45 Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996. `doi:10.1137/1038003`.

46 Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1101–1113, 1993.