

A SPECTRAL ALGORITHM FOR IMPROVING GRAPH PARTITIONS

Michael W. Mahoney*

Lorenzo Orecchia[†]

Nisheeth K. Vishnoi[‡]

December 3, 2009

Abstract

In the cut-improvement problem, we are asked, given a starting cut (T, \bar{T}) in a graph G , to find a cut with low conductance *around* (T, \bar{T}) or produce a certificate that there is none. More precisely, for a notion of correlation between cuts, cut-improvement algorithms seek to produce approximation guarantees of the following form: for any cut (C, \bar{C}) that is ε -correlated with the input cut, the cut output by the algorithm has conductance upper-bounded by a function of the conductance of (C, \bar{C}) and ε .

In this paper we approach the cut-improvement problem from a spectral perspective: given a graph G and a cut (T, \bar{T}) , we first associate a natural unit vector to (T, \bar{T}) . Then we modify the standard spectral relaxation for G to bias it towards this vector, and use this relaxation to present a new cut-improvement algorithm, `SPIMPROVE`. Our relaxation gives rise to a geometric notion of correlation among cuts. Moreover, we show that the classic sweep-cut rounding can still be applied and that we can solve our relaxation in nearly linear time by reducing it to an eigenvector computation.

Further, we show how our approach is intimately connected to electrical networks in one limiting case. We believe this connection may be of independent interest. Finally, we compare our algorithm to the previous flow-based algorithm of Andersen and Lang [AL08]. We show that `SPIMPROVE` is likely to be faster and give better approximation guarantees in many settings.

*Stanford University. Email: mahoneymw@gmail.com

[†]UC Berkeley. Email: orecchia@eecs.berkeley.edu

[‡]Microsoft Research India, Bangalore. Email: nisheeth.vishnoi@gmail.com

1 Introduction

Graph partitioning and conductance. A fundamental problem in graph partitioning is the minimum-conductance-cut problem which is defined as follows: given a graph $G = (V, E)$, find a partition (S, \bar{S}) of the vertices that has the minimum conductance. The conductance of a cut (S, \bar{S}) is defined to be $\frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$ where $\text{vol}(S)$ is the sum of the degrees of the vertices in the set S .

Interest in this problem derives both from its numerous practical applications, such as image segmentation, VLSI layout and clustering (see the survey of Shmoys [Shm97]), and from its theoretical connections to areas such as random walks, spectral methods and semidefinite programming. The minimum-conductance-cut problem is NP-hard and the best approximation algorithm known for it is due to Arora, Rao and Vazirani (ARV) [ARV04], who establish an $O(\sqrt{\log n})$ factor algorithm for this problem using techniques from semidefinite programming and measure concentration.

The main challenge in designing applied algorithms for graph partitioning is to combine quality with scalability. As typical inputs to this problem consist of very large graphs, it is imperative to find algorithms that not only provide a guarantee about the quality of the cut they produce, but also have fast running times. In a recent breakthrough, Sherman [She09], following up on the line of work by [KRV06, OSVV08, AK07] shows how to achieve $O(\sqrt{\log n})$ approximation using a polylogarithmic number of single-commodity maximum-flow computations.

A parallel line of research has focused on the application of spectral methods to graph partitioning [AM85, KVV00] and has culminated with the work of Spielman and Teng on nearly-linear-time algorithms [ST03]. These methods are not "true" approximation algorithms, as their approximation ratios depend on the objective value, but they display many favorable properties, including faster running times in practice to better approximation guarantees on certain families of graphs.

Cut-improvement algorithms. Many of the fastest algorithms for finding minimum-conductance cuts, such as those in [KRV06, OSVV08, AK07, She09], use as a crucial building block a primitive that takes as input a cut (T, \bar{T}) and attempts to find a lower-conductance cut that is *well correlated* with (T, \bar{T}) . This primitive is referred to as a *cut-improvement algorithm* [LR04, AL08] as its original purpose was limited to post-processing cuts output by other algorithms. Recently, cut-improvement algorithms have found more application, for instance they have been used to find low conductance in specific regions of large graphs [LLDM08].

Given a notion of correlation between cuts, cut-improvement algorithms typically produce approximation guarantees of the following form: for any cut (C, \bar{C}) that is ε -correlated with the input cut, the cut output by the algorithm has conductance upper-bounded by a function of the conductance of (C, \bar{C}) and ε .

Previous Work. Gallo *et al.* [GGT89] were the first to show that one can find a subset of an input set $T \subseteq V$ with minimum conductance in polynomial time. Lang and Rao [LR04] implement a closely related algorithm and demonstrate its effectiveness at refining cuts output by other methods. This line of research culminates in the work of Andersen and Lang [AL08], who give a more general algorithm that uses a small number of single-commodity maximum-flows to find low-conductance cuts not only inside the input subset T , but among all cuts which are well-correlated with (T, \bar{T}) .

This paper. In this paper we approach the cut-improvement problem from a spectral perspective in contrast with the previous flow-based algorithms. Given a graph G and a cut (T, \bar{T}) in it, we first associate a unit vector corresponding to this cut. This vector is orthogonal to the all 1s vector. We then write down a modified version of the standard spectral problem for G . This new program takes this unit vector, and a *bias* parameter β , and replaces the global variance term appearing at the denominator in the standard program with the square of a linear function, which can be interpreted as a measure of correlation with the unit vector corresponding to (T, \bar{T}) . The parameter

$\beta \in [0, 1)$, with 0 being the standard spectral relaxation while the limiting case of $\beta = 1$ reduces to a electrical network problem. Next, we show that this modified program behaves similarly to the standard spectral relaxation: its objective value can be used to lower bound the conductance of cuts and the optimal vector can be rounded by picking the best sweep cut. The resulting approximation guarantee depends on a new measure of correlation between subsets of vertices. We are able to solve our mathematical program in nearly linear time by reducing it to an eigenvector computation.

Comparison. Our use of spectral techniques is also inspired by the observation that, in many graph partitioning problems, spectral and flow techniques display complementary strengths and weaknesses. For example, spectral methods achieve good approximations on expanders and expander-like graphs where flow methods can perform poorly, while the opposite holds for graphs with long paths and very low-conductance cuts [Chu97]. We argue that our method may possess two advantages over the flow-based method of Andersen and Lang [AL08]. First, the running time of our algorithm is dominated by an eigenvalue computation which, for many graphs, and in particular for expander-like graphs, is faster than that of a maximum flow computation. This advantage is amplified in practice as many eigenvector solvers usually perform much better than their worst-case guarantees and have had their code thoroughly optimized. Secondly, comparing Theorem 1.1 with the main theorem of Andersen and Lang [AL08], we show that our algorithm gives a better approximation guarantee for many graphs and cuts, including for many cuts within expander-like graphs and in the case that the volume of the input cut is significantly smaller than that of the cut which we are interested in bounding. In this sense, our algorithm complements the flow-based approach of [AL08], which performs best when the target cut is a subset of the input cut.

At this time, we have not yet run experiments to validate these observations, but we believe they may hold based on their consistency with the known behavior of spectral and flow methods and on previous empirical evaluations of the flow-based algorithms [LMO09].

Future work. We believe that our mathematical programming approach and the connection to electrical networks may be of independent interest. Indeed, very recently, this idea of using a strong mathematical programming relaxation and coming up with a nearly-linear time algorithm to solve it, has led to the best nearly-linear time algorithm for graph partitioning [OV09]. We anticipate more applications in the graph partitioning domain in the near future.

1.1 Our Results

Preliminaries. We assume that the instance graph $G = (V, E)$ is undirected, unweighted and connected with $|V| = n$ and $|E| = m$. All our results apply for weighted graphs. For $i \in V$, we denote by d_i the degree of vertex i in G and, for a subset $S \subseteq V$, we define $\text{vol}(S) := \sum_{i \in S} d_i$. Notice that $\text{vol}(G) := \text{vol}(V) = 2m$. The conductance of a cut (S, \bar{S}) in G is defined as:

$$\phi(S) := \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}.$$

The conductance of the graph is referred to as $\phi(G)$ and is the minimum of $\phi(S)$ over all subsets $S \subseteq V$. We will call a graph an *expander* if $\phi(G) = \Omega(1)$.

Correlation between cuts. We consider the following measure of correlation between a cut (A, \bar{A}) and a cut (B, \bar{B}) :

$$K(A, B) := \frac{\text{vol}(B)\text{vol}(\bar{B})}{\text{vol}(A)\text{vol}(\bar{A})} \left(\frac{\text{vol}(A \cap B)}{\text{vol}(B)} - \frac{\text{vol}(A \cap \bar{B})}{\text{vol}(\bar{B})} \right)^2.$$

It can be proved that K must lie in the interval $[0, 1]$. The correlation function naturally arises from the spectral relaxation we consider and is discussed in Section 1.2. Several properties of this correlation are listed in Appendix A for reference.

Main Theorem. The following is the main theorem of this paper.

Theorem 1.1. *Let $\beta \in [0, 1)$ be a parameter and let $\varepsilon \in (0, 1)$. For all (C, \bar{C}) such that $K(C, T) \geq \varepsilon$, $\text{SPIMPROVE}(\beta, T)$ outputs a cut (S, \bar{S}) such that*

$$\phi(S) \leq 2 \sqrt{\frac{\phi(C)}{1 - \beta + \beta\varepsilon}}.$$

The running time of this algorithm is $\tilde{O}(m/\phi^2(S))$.

The parameter β establishes the relative importance of approximating cuts around the input cut. For $\beta = 1 - \Omega(1)$, our result is just a weaker form of the classic spectral partitioning result of [AM85], but for higher values of β , our algorithm gives bounds focused on cuts around the input cut and less dependent of the global structure. This can also contribute to reduce the running time, if the cuts correlated with the input cut have higher conductance than the global minimum-conductance cut in the graph.

Electrical Flows and $\beta = 1$. For $\beta = 1$, we can prove a similar result using connection to electrical networks. The details are presented in Section 5.

1.2 Overview and Techniques

Now we outline our techniques and present the SPIMPROVE algorithm. We will use a slightly different notion of conductance

$$\Phi(S) = \frac{|E(S, \bar{S})|}{\text{vol}(S)\text{vol}(\bar{S})}$$

and denote by Φ its minimum over subsets $S \subseteq V$. Notice that for any $S \subseteq V$, $\text{vol}(G) \cdot \Phi(S) \geq \phi(S) \geq \frac{\text{vol}(G)}{2} \cdot \Phi(S)$. Hence, Φ is only a constant factor away from a scaling of conductance. Our analysis relies on minimizing a modified objective function that optimizes a trade-off of small conductance and large correlation with (T, \bar{T}) . The trade-off is regulated by a parameter $\beta \in [0, 1]$. The precise quantity we consider is the following:

$$\Phi_{\beta, T}(S) := \frac{|E(S, \bar{S})|}{\text{vol}(S)\text{vol}(\bar{S})((1 - \beta) + \beta K(S, T))} = \frac{\Phi(S)}{(1 - \beta) + \beta K(T, S)}.$$

Denote by $\Phi_{\beta, T}$ the minimum of $\Phi_{\beta, T}(S)$ over all subsets $S \subseteq V$. Notice that $\Phi_{0, T}$ is just Φ , while for $\beta > 0$, the cuts S with $K(S, T) = c$ have their objective value increased by a factor of $((1 - \beta) + \beta c)^{-1} > 1$. Hence, $\Phi_{\beta, T}$ will penalize cuts that are poorly correlated with (T, \bar{T}) and bias the minimization towards cuts which are well correlated with (T, \bar{T}) . Moreover, it is easy to check that $\frac{\Phi(S)}{1 - \beta} \geq \Phi_{\beta, T}(S) \geq \Phi(S)$. To approximate $\Phi_{\beta, T}$ we proceed to write a spectral relaxation of it similar to that for Φ .

A Relaxation. Consider the following vector optimization program and let $\lambda_{\beta, T}$ be its optimal value.

$$\text{PROG : } \min_{x \in \mathbb{R}^V} \frac{\sum_{i, j \in E(G)} (x_i - x_j)^2}{(1 - \beta) \sum_{i < j} d_i d_j (x_i - x_j)^2 + \beta \text{vol}(T) \text{vol}(\bar{T}) \left(\frac{\sum_{i \in T} d_i x_i}{\text{vol}(T)} - \frac{\sum_{i \in \bar{T}} d_i x_i}{\text{vol}(\bar{T})} \right)^2}$$

Motivation for our correlation function. For a cut (T, \bar{T}) , define a representative the vector s_T by letting $s_T(i) = \sqrt{\text{vol}(T)\text{vol}(\bar{T})/2m} \cdot 1/\text{vol}(T)$ if $i \in T$ and $s_T(i) = -\sqrt{\text{vol}(T)\text{vol}(\bar{T})/2m} \cdot 1/\text{vol}(\bar{T})$. Then s_T is a unit vector, i.e. $\langle s_T, Ds_T \rangle$, and $\langle s_T, D1 \rangle = 0$. We claim that the term in the denominator with coefficient β measures the correlation of the vector x with s_T and has a natural geometric interpretation. This motivates the form of our correlation K , as it can be shown that , for a set (U, \bar{U}) ,

$$K(S, T) = \langle s_T, Ds_U \rangle^2.$$

In Section 4.1, we will analyze this program and show that it is a relaxation for $\Phi_{\beta, T}$:

Lemma 1.2. $\lambda_{\beta, T} \leq \Phi_{\beta, T}$.

The algorithm. Our algorithm SPIMPROVE is described in Figure 1. The algorithm SPIMPROVE on input (β, T) solves the relaxation PROG of Figure 5. Given the optimal vector x^* , SPIMPROVE outputs the minimum conductance cut (S, \bar{S}) among the *sweep cuts* of x^* .

1. **INPUT:** A graph $G(V, E)$, $\beta \in [0, 1)$ and a cut (T, \bar{T}) .
2. Compute x^* , the optimal to PROG on this instance (see Lemma 1.4 below).
3. Sort the entries of the vector x^* . Without loss of generality assume that

$$x_1^* \geq x_2^* \geq \dots \geq x_n^*.$$

4. Let (S_i, \bar{S}_i) denote the cut $(\{1, 2, \dots, i\}, \{i+1, \dots, n\})$ for $1 \leq i \leq n-1$.
5. **OUTPUT:** The cut (S, \bar{S}) which minimizes $\phi(S_i)$ for $1 \leq i \leq n-1$.

Figure 1: Algorithm SPIMPROVE

Rounding and Approximation. The proof of Theorem 1.1 relies on the following lemma, which is a consequence of the sweep-cut rounding performed by SPIMPROVE:

Lemma 1.3. SPIMPROVE (β, T) outputs a cut (S, \bar{S}) such that

$$\phi(S) \leq \sqrt{2 \cdot \text{vol}(G) \cdot \lambda_{\beta, T}}.$$

Notice that our final guarantee is on the conductance of the output cut (S, \bar{S}) and that we make no claim about $\Phi_{\beta, T}(S)$. This lemma is proved in Section 4.2.

Running Time. In Section 5, we describe how the optimal solution to PROG can be computed in time $O(1/\lambda_{\beta, T})$, by decomposing the denominator to reduce the problem to an eigenvector computation and appealing to a standard method, such as the Power Method [GVL96].

Lemma 1.4. For every $\beta \in [0, 1)$ and $T \subseteq V$, the optimal solution to PROG can be computed in time $O(1/\lambda_{\beta, T})$.

Proof of Main Theorem. Before proving Lemmata 1.2, 1.3 and 1.4, we show how they imply Theorem 1.1.

Proof. Let (C, \bar{C}) be a cut such that $K(C, T) = \varepsilon$. Then

$$\Phi_{\beta, T} \leq \Phi_{\beta, T}(C) = \frac{\Phi(C)}{1 - \beta + \beta\varepsilon}.$$

Combining this with Lemma 1.2 and Lemma 1.3 yields:

$$\phi(S) \leq \sqrt{2 \cdot \text{vol}(G) \cdot \lambda_{\beta, T}} \leq \sqrt{2 \cdot \text{vol}(G) \cdot \Phi_{\beta, T}} \leq \sqrt{2 \cdot \text{vol}(G) \cdot \frac{\Phi(C)}{1 - \beta + \beta\varepsilon}} \leq 2 \sqrt{\frac{\phi(C)}{1 - \beta + \beta\varepsilon}}.$$

The last inequality follows as $1/2 \cdot \text{vol}(G) \cdot \Phi(C) \leq \phi(C)$.

Furthermore, we can combine Lemma 1.4 and 1.3 to prove that the running time of computing a solution to PROG is at most $O(m/\phi^2(S))$. As finding the minimum-conductance sweep cut takes time $\tilde{O}(m)$, the total running time is at most $\tilde{O}(m/\phi^2(S))$. \square

Rest of the paper. In Section 2 we give a detailed comparison of our work with that of Andersen and Lang [AL08]. In Section 3 we introduce notation and basic facts, while in Section 4 we present the proofs of the main results. We conclude with a brief discussion on the interpretation of our relaxation and on possible extensions in Section ??.

2 Comparison with Previous Algorithms

2.1 The correlation function

In this subsection, we compare our correlation function K with the correlation function F implicit in the flow-based cut-improvement algorithm of Andersen and Lang [AL08]. Given two cuts (C, \bar{C}) and (T, \bar{T}) such that $\text{vol}(C) \leq \text{vol}(\bar{C})$ and $\text{vol}(T) \leq \text{vol}(\bar{T})$, their correlation $F(C, T)$ is defined as

$$F(C, T) := \frac{\text{vol}(T)}{\text{vol}(C)} \left(\frac{\text{vol}(C \cap T)}{\text{vol}(T)} - \frac{\text{vol}(C \cap \bar{T})}{\text{vol}(\bar{T})} \right).$$

If $F(C, T) = \varepsilon$, it is possible to show that

$$\frac{\text{vol}(C \cap T)}{\text{vol}(C)} = \frac{\text{vol}(T)}{\text{vol}(G)} + \varepsilon \frac{\text{vol}(\bar{T})}{\text{vol}(G)}.$$

Hence, we can interpret the correlation function F as a measure of how much larger the fraction of C contained in T is with respect to what one would expect for a random C , i.e., $\frac{\text{vol}(T)}{2m}$ [AL08]. The next lemma shows that the correlation measures K and F are tightly related.

Lemma 2.1. *For any cuts (C, \bar{C}) and (T, \bar{T}) such that $\text{vol}(C) \leq \text{vol}(\bar{C})$ and $\text{vol}(T) \leq \text{vol}(\bar{T})$, we have:*

$$\frac{\text{vol}(T)}{\text{vol}(C)} K(C, T) \leq F(C, T)^2 \leq \frac{2\text{vol}(T)}{\text{vol}(C)} K(C, T).$$

The proof of this simple lemma is omitted.

2.2 Spectral vs. Flow

In this subsection, we compare the approximation guarantee of SpIMPROVE with that of the flow-based algorithm of Andersen and Lang [AL08]. We start by recalling the main theorem of [AL08].

Theorem 2.2. [AL08] *On input (T, \bar{T}) with $\text{vol}(T) \leq \text{vol}(\bar{T})$, the flow-based cut-improvement algorithm outputs a cut (S, \bar{S}) such that, for all cuts (C, \bar{C}) with $\text{vol}(C) \leq \text{vol}(\bar{C})$, we have*

$$\phi(S) \leq \frac{\phi(C)}{F(C, T)}.$$

The next theorem uses Lemma 2.1 to restate the approximation guarantee of Theorem 1.1 as a function of $F(C, T)$ rather than $K(C, T)$. Its statement can be compared directly to the main theorem in Andersen and Lang [AL08].

Theorem 2.3. *For all $\beta \in (0, 1]$, on input (T, \bar{T}) with $\text{vol}(T) \leq \text{vol}(\bar{T})$, SpIMPROVE outputs a cut (S, \bar{S}) such that, for all cuts (C, \bar{C}) with $\text{vol}(C) \leq \text{vol}(\bar{C})$, we have*

$$\phi(S) \leq \frac{2\sqrt{\phi(\bar{C})}}{F(C, T)} \sqrt{\frac{2}{\beta} \cdot \frac{\text{vol}(T)}{\text{vol}(C)}}.$$

Proof. Consider the statement of Theorem 1.1 and use the fact that $K(C, T) \geq 1/2 \cdot \frac{\text{vol}(C)}{\text{vol}(T)} F(C, T)^2$ from Lemma 2.1.

$$\phi(S) \leq 2 \sqrt{\frac{\phi(C)}{1 - \beta + \beta K(C, T)}} \leq 2 \sqrt{\frac{\phi(C)}{\beta K(C, T)}} \leq \frac{2\sqrt{\phi(\bar{C})}}{F(C, T)} \sqrt{\frac{2}{\beta} \cdot \frac{\text{vol}(T)}{\text{vol}(C)}}.$$

□

Comparing the statements of Theorem 1.1 and Theorem 2.2, we can compute the ratio of the approximation guarantee of SpIMPROVE to that of [AL08] as

$$\frac{2}{\sqrt{\phi(\bar{C})}} \sqrt{\frac{2}{\beta} \cdot \frac{\text{vol}(T)}{\text{vol}(C)}}. \quad (1)$$

Note that we could have obtained a tighter, albeit more complicated, form for this ratio, which is optimal only for $\beta = 1$. However, the current form will suffice for the purposes of this section. We now make two observations about the relative performance of the two algorithms. First, notice that the approximation of SpIMPROVE is relatively better when $\phi(C)$ is large, e.g., when G is an expander, and that this is consistent with our expectation that a spectral method would be more suitable in such cases. Secondly, (1) shows that, if the input cut has volume sufficiently smaller than that of C (the cut we are interested in bounding) the guarantee given by SpIMPROVE is stronger than that of the flow-based method. Indeed, this suggests our spectral method may be complementary to the flow-based approach, which performs best when the target cut is contained in the input cut. We conclude this comparison by noticing that when G is an expander or expander-like, not only SpIMPROVE has a performance guarantee close or better to that of [AL08], it also has an almost linear running time by Theorem 1.4, potentially much faster than that of the flow algorithm, which can only be upper bounded by $\tilde{O}(m^{3/2})$ [GR98].

3 Preliminaries

3.1 Definitions

Graph matrices. For an undirected graph $H = (V, E_H)$, let $A(H)$ denote the adjacency matrix of H and $D(H)$ the diagonal matrix of degrees of H . The (combinatorial) Laplacian of H is defined as $L(H) := D(H) - A(H)$. $L(H)$ has the property that for all $x \in \mathbb{R}^V$, $x^T L(H)x = \sum_{ij \in E_H} (x_i - x_j)^2$. By A, D and L , we denote $A(G), D(G)$ and $L(G)$ respectively. For a symmetric matrix M , we will use $M \geq 0$ to denote that it is positive semi-definite and $M > 0$ to denote that it is positive definite. The expression $A \geq B$ is equivalent to $A - B \geq 0$.

Vectors. For a subset S of vertices, we denote by 1_S the indicator vector of S in \mathbb{R}^V . Also, we denote by 1 the vector in \mathbb{R}^V having all entries set to 1. We consider the following cuts that arise from vectors.

Definition 3.1 (Sweep Cut). Given a vector $x \in \mathbb{R}^V$ a sweep cut of it is a cut of the form $(\{x_1, \dots, x_k\}, \{x_{k+1}, \dots, x_n\})$ for some k , such that $x_1 \geq x_2 \geq \dots \geq x_k \geq x_{k+1} \geq \dots \geq x_n$.

Special matrices. Recall D is the diagonal matrix of degrees of the vertices of the instance graph G . We define the complete graph K_n on V by its adjacency matrix.

$$A(K_n) := DJD = D11^T D$$

Notice that this is not the standard complete graph, but a weighted version of it, where the weights depend on D . Indeed, with this scaling we have $D(K_n) = 2mD = \text{vol}(G)D$. Hence, the Laplacian of the complete graph defined in this manner becomes

$$L(K_n) = \text{vol}(G)D - DJD.$$

Given a subset $T \subseteq V$, we also define:

$$s_T := \sqrt{\text{vol}(T)\text{vol}(\bar{T})} \left(\frac{1_T}{\text{vol}(T)} - \frac{1_{\bar{T}}}{\text{vol}(\bar{T})} \right).$$

Finally, define $\Omega_T := Ds_T(s_T)^T D$. Notice that $\Omega_T = \Omega_{\bar{T}}$.

3.2 Basic Facts

Here we introduce some simple observations which will be used in the main proofs.

Fact 3.2. $s_T^T D1 = 0$.

Proof. It follows from the observation that

$$\left(\frac{1_T}{\text{vol}(T)} - \frac{1_{\bar{T}}}{\text{vol}(\bar{T})} \right)^T D1 = \frac{\text{vol}(T)}{\text{vol}(T)} - \frac{\text{vol}(\bar{T})}{\text{vol}(\bar{T})} = 1 - 1 = 0.$$

□

Fact 3.3. $s_T^T Ds_T = \text{vol}(G)$.

Proof. It follows from the observation that

$$s_T^T Ds_T = \text{vol}(T)\text{vol}(\bar{T}) \left(\frac{\text{vol}(T)}{\text{vol}(T)^2} + \frac{\text{vol}(\bar{T})}{\text{vol}(\bar{T})^2} \right) = \text{vol}(T) + \text{vol}(\bar{T}) = \text{vol}(G).$$

□

Fact 3.4. $L(K_n) \geq \Omega_T$

Proof. Notice that $L(K_n) = \text{vol}(G)D - DJD$, while $\Omega_T = Ds_T(Ds_T)^T$. Since $L(K_n) \geq 0$ and Ω_T is rank one, it suffices to prove that

$$(s_T)^T (\text{vol}(G)D - D11^T D) s_T \geq (s_T)^T Ds_T (s_T)^T Ds_T.$$

By Fact 3.3, the first term in the l.h.s. above becomes $\text{vol}(G) \cdot (s_T)^T Ds_T = \text{vol}(G)^2$. By Fact 3.2, the second term in the l.h.s. is 0. Also, Fact 3.3 implies that the r.h.s. is $\text{vol}(G)^2$. Hence, this fact is proved. □

Next we state a standard result on rounding spectral relaxations of graph partitioning problems. The proof of this is identical to Theorem 2.2 in Chapter 2 of [Chu97] and we choose not to repeat it in this version of the paper.

Theorem 3.5 (Sweep-Cut Rounding). *Given a vector x such that $\lambda = \frac{x^T Lx}{x^T L(K_n)x}$, there exists a sweep cut S of x such that*

$$\phi(S) \leq \sqrt{2 \cdot \text{vol}(G) \cdot \lambda}.$$

4 Proofs

In this section we prove Lemmata 1.2, 1.3 and 1.4, which we have shown to combine to yield Theorem 1.1.

4.1 The Relaxation.

We start by expressing the relaxation PROG using the notation of Section 3. Notice that, for all $x \in \mathbb{R}^V$:

$$x^T \Omega_T x = (x^T D_{ST})^2 = \text{vol}(T) \text{vol}(\bar{T}) \left(\frac{\sum_{i \in T} d_i x_i}{\text{vol}(T)} - \frac{\sum_{i \in \bar{T}} d_i x_i}{\text{vol}(\bar{T})} \right)^2.$$

Hence, we can rewrite PROG as:

$$\text{PROG : } \min_{x \in \mathbb{R}^V} \frac{x^T L x}{x^T ((1 - \beta)L(K_n) + \beta \Omega_T) x}$$

We now prove Lemma 1.2:

Proof. For any $x \in \mathbb{R}^V$, we have:

$$\lambda_{\beta, T} \leq \frac{x^T L x}{x^T ((1 - \beta)L(K_n) + \beta \Omega_T) x} = \frac{x^T L x}{x^T L(K_n) x \left((1 - \beta) + \beta \frac{x^T \Omega_T x}{x^T L(K_n) x} \right)}. \quad (2)$$

Notice how the denominator has now a similar form to that of $\Phi_{\beta, T}$. Consider now the subset S which minimizes the objective $\Phi_{\beta, T}(S)$, so that $\Phi_{\beta, T} = \Phi_{\beta, T}(S)$ and set $x = 1_S$ in (2). The following equalities hold:

$$\begin{aligned} 1_S^T L 1_S &= \sum_{i, j \in E} (1_S(i) - 1_S(j))^2 = |E(S, \bar{S})|, \\ 1_S^T L(K_n) 1_S &= \sum_{i < j} d_i d_j (1_S(i) - 1_S(j))^2 = \text{vol}(S) \text{vol}(\bar{S}), \\ 1_S^T \Omega_T 1_S &= (1_S^T D_{ST})^2 = \text{vol}(T) \text{vol}(\bar{T}) \left(\frac{\text{vol}(S \cap T)}{\text{vol}(T)} - \frac{\text{vol}(S \cap \bar{T})}{\text{vol}(\bar{T})} \right)^2. \end{aligned}$$

This implies that $\frac{1_S^T \Omega_T 1_S}{1_S^T L(K_n) 1_S} = K(S, T)$. Hence, we can now complete the proof:

$$\lambda_{\beta, T} \leq \frac{|E(S, \bar{S})|}{\text{vol}(S) \text{vol}(\bar{S}) ((1 - \beta) + \beta K(S, T))} = \Phi_{\beta, T}(S) = \Phi_{\beta, T}.$$

□

4.2 Rounding and Approximation

In this section we prove Lemma 1.3. To do so, we relate $\lambda_{\beta, T}$ to a ratio of the same form as that in Theorem 3.5.

Proof. Let x^* be a vector which achieves value $\lambda_{\beta, T}$ in the program PROG. By Fact 3.4:

$$\lambda_{\beta, T} = \frac{x^{*T} L x^*}{x^{*T} ((1 - \beta)L(K_n) + \beta \Omega_T) x^*} \geq \frac{x^{*T} L x^*}{x^{*T} L(K_n) x^*}.$$

Now, from Theorem 3.5, it follows that there is a sweep cut (S, \bar{S}) of the vector x^* such that

$$\phi(S) \leq \sqrt{2 \cdot \text{vol}(G) \cdot \frac{x^{*T} L x^*}{x^{*T} L(K_n) x^*}} \leq \sqrt{2 \cdot \text{vol}(G) \cdot \lambda_{\beta, T}}. \quad (3)$$

This proves Lemma 1.3. □

4.3 Computing the Optimal of the Spectral Relaxation

In this section we prove Lemma 1.4. The key idea is to rephrase the optimization problem PROG as an eigenvector problem and then appeal to the standard Power Method (see [GVL96]) to compute the smallest eigenvector of a matrix. Hence, we focus on reducing this optimization problem to an eigenvector problem. Recall the program:

$$\text{PROG} : \min_{x \in \mathbb{R}^V} \frac{x^T Lx}{x^T [(1 - \beta)(\text{vol}(G)D - DJD) + \beta D s_T (s_T)^T D] x},$$

where

$$s_T := \sqrt{\text{vol}(T)\text{vol}(\bar{T})} \left(\frac{1_T}{\text{vol}(T)} - \frac{1_{\bar{T}}}{\text{vol}(\bar{T})} \right).$$

Let $\mathcal{L} := D^{-1/2} L D^{-1/2}$ be the normalized Laplacian of G . Since G is connected $D > 0$. Hence, we can substitute $y = D^{1/2} x$ in the minimization problem above to obtain

$$\text{PROG} : \min_{y \in \mathbb{R}^V} \frac{y^T \mathcal{L} y}{y^T [(1 - \beta)(\text{vol}(G)I - D^{1/2} J D^{1/2}) + \beta D^{1/2} s_T (s_T)^T D^{1/2}] y}.$$

Finally, let the matrix \mathcal{M} be defined as follows:

$$\mathcal{M} := (1 - \beta) \left(I - \frac{D^{1/2} J D^{1/2}}{\text{vol}(G)} \right) + \beta \left(\frac{D^{1/2} s_T (s_T)^T D^{1/2}}{\text{vol}(G)} \right).$$

We can then rewrite PROG as follows

$$\text{PROG} : \frac{1}{\text{vol}(G)} \cdot \min_{y \in \mathbb{R}^V} \frac{y^T \mathcal{L} y}{y^T \mathcal{M} y}$$

The next step is to characterize the eigenspaces of the matrix \mathcal{M} in the denominator. This will allow us to write down the transformation necessary to write PROG as an eigenvector problem and will also be the basis for our interpretation of the relaxation PROG in Section ??.

Lemma 4.1. *Let $\beta \in [0, 1)$. Then the eigenspaces of the matrix \mathcal{M} are*

1. *The span of vector $D^{1/2} \mathbf{1}$ with eigenvalue 0.*
2. *The span of vector $D^{1/2} s_T$ with eigenvalue 1.*
3. *The subspace \mathcal{T} orthogonal to the span of $D^{1/2} \mathbf{1}$ and $D^{1/2} s_T$ with eigenvalue $(1 - \beta)$.*

Proof. Using Fact 3.3 and the fact that $\mathbf{1}^T D \mathbf{1} = \text{vol}(G)$, we can rewrite \mathcal{M} as:

$$\mathcal{M} = (1 - \beta) \cdot I - (1 - \beta) \cdot \left(\frac{D^{1/2} \mathbf{1}}{\|D^{1/2} \mathbf{1}\|_2} \right) \left(\frac{D^{1/2} \mathbf{1}}{\|D^{1/2} \mathbf{1}\|_2} \right)^T + \beta \cdot \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right) \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right)^T.$$

Notice that the matrices in second and third term of this sum are projection matrices over the unit vectors in the direction of $D^{1/2} \mathbf{1}$ and $D^{1/2} s_T$ respectively. Notice also that these two vectors are orthogonal by Fact 3.2. Hence:

1. $\mathcal{M}(D^{1/2} \mathbf{1}) = (1 - \beta) \cdot (D^{1/2} \mathbf{1}) - (1 - \beta) \cdot (D^{1/2} \mathbf{1}) = 0$,
2. $\mathcal{M}(D^{1/2} s_T) = (1 - \beta) \cdot (D^{1/2} s_T) + \beta (D^{1/2} s_T) = 1 \cdot (D^{1/2} s_T)$
3. For any $x \in \mathcal{T}$: $\mathcal{M}x = (1 - \beta) \cdot x$.

This completes the proof of Lemma 4.1. □

Resuming the proof of Theorem 1.4, we note that if $\beta \in [0, 1)$, we can define the square root $\mathcal{M}^{-1/2}$ of the pseudo-inverse of matrix \mathcal{M} . Now, the eigenvector decomposition in Lemma 4.1 immediately allows us to write:

$$\begin{aligned}\mathcal{M}^{-1/2} &= 1 \cdot \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right) \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right)^T + \sqrt{\frac{1}{1-\beta}} \cdot \left[I - \left(\frac{D^{1/2} 1}{\|D^{1/2} 1\|_2} \right) \left(\frac{D^{1/2} 1}{\|D^{1/2} 1\|_2} \right)^T - \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right) \left(\frac{D^{1/2} s_T}{\|D^{1/2} s_T\|_2} \right)^T \right] \\ &= \sqrt{\frac{1}{1-\beta}} \cdot I - \sqrt{\frac{1}{1-\beta}} \cdot \frac{D^{1/2} J D^{1/2}}{\text{vol}(G)} + \left(1 - \sqrt{\frac{1}{1-\beta}} \right) \cdot \frac{D^{1/2} s_T (s_T)^T D^{1/2}}{\text{vol}(G)}.\end{aligned}$$

Hence, we can use the transformation $y = \mathcal{M}^{-1/2} z$ to finally obtain:

$$\text{PROG} : \frac{1}{\text{vol}(G)} \cdot \min_{z \in \mathbb{R}^V} \frac{z^T \mathcal{M}^{-1/2} \mathcal{L} \mathcal{M}^{-1/2} z}{z^T z}.$$

This shows that PROG can be stated as a standard eigenvector problem, i.e. the problem of determining the smallest eigenvalue σ and corresponding eigenvector of the matrix $\mathcal{M}^{-1/2} \mathcal{L} \mathcal{M}^{-1/2}$. The optimal value $\lambda_{\beta, T}$ of PROG is then equal to $\sigma/\text{vol}(G)$.

Moreover, as $\mathcal{M}^{-1/2}$ has eigenvalues between 0 and 1 and \mathcal{L} has eigenvalues between 0 and 2, it must be the case that the eigenvalues of $\mathcal{M}^{-1/2} \mathcal{L} \mathcal{M}^{-1/2}$ are contained in $[0, 2]$. Hence, we can use the Power Iteration outlined in [GVL96] to produce a vector of objective value only an arbitrarily small constant larger than $\lambda_{\beta, T}$ in $O\left(\frac{1}{\sigma}\right) = O\left(\frac{1}{\text{vol}(G) \cdot \lambda_{\beta, T}}\right)$ iterations. Each iteration takes time proportional to the time of a multiplying a vector by matrix $\mathcal{M}^{-1/2} \mathcal{L} \mathcal{M}^{-1/2}$. However, this can be done in time $O(m)$ as \mathcal{L} is sparse with $O(m)$ non-zero entries and $\mathcal{M}^{-1/2}$ is a linear combination of the identity and two rank-one matrices, which can be multiplied with a vector in time $O(n)$. This gives a total running time of $O\left(\frac{1}{\lambda_{\beta, T}}\right)$, as required by Lemma 1.4.

5 Connection to Electrical Networks

In this section we study the limiting case of $\beta = 1$. This requires special treatment as the relaxation has to be solved via methods different than eigenvalue type methods. More interestingly, the relaxation corresponds to setting up an electrical flow problem in the graph G around the input cut (T, \bar{T}) . We believe this is of independent interest. We keep this section brief in this version of the paper and omit most of the details. We first state the main result in the $\beta = 1$ case.

Theorem 5.1. *Given an undirected graph $G = (V, E)$, an error parameter $\varepsilon > 0$ and an input subset $T \subseteq V$, let (S, \bar{S}) be the cut output by $\text{SPIMPROVE}(1, T)$. Then, for all (C, \bar{C}) ,*

$$\phi(S) \leq (1 + \varepsilon) \cdot \sqrt{2 \cdot \frac{\phi(C)}{K(C, T)}}.$$

The running time is $\tilde{O}(m \cdot \log(1/\varepsilon))$ by usage of the Spielman-Teng solver of [ST03].

Relaxation. Recall that for $\beta = 1$, and the cut (T, \bar{T}) , the program PROG can be written as

$$\text{PROG}' : \min_{x \in \mathbb{R}^V} \frac{x^T L x}{\langle x, D s \rangle^2} = \frac{\sum_{i,j \in E} (x_i - x_j)^2}{\left(\sum_{i \in V} d_i s_i x_i \right)^2},$$

where $s \in \mathbb{R}^V$ is the vector $s(i) = \sqrt{\text{vol}(\bar{T})/\text{vol}(T)}$ if $i \in T$ and $s(i) = -\sqrt{\text{vol}(T)/\text{vol}(\bar{T})}$ if $i \in \bar{T}$. Let λ_T denote the optimal value of this program.

Algorithm. We can still use SPIMPROVE but since the form of relaxation is different, we will have to devise a new method of solving it. We will show that the program above can be reduced to solving the system of linear equations $Lx = Ds$ and we will present a method to compute an approximately optimal solution, achieving the following guarantee.

Lemma 5.2. *A feasible solution to PROG' of value at most $(1 + \varepsilon) \cdot \lambda_T$ can be computed in time $\tilde{O}(m \log(1/\varepsilon))$ using the Spielman-Teng linear-equation solver [ST03].*

Computing the optimal. Let L^+ denote the pseudo-inverse of L .

Lemma 5.3. *For any s such that $\langle s, D1 \rangle = 0$, $\frac{x^T Lx}{(x^T Ds)^2} \geq \frac{1}{s^T D L^+ Ds}$.*

Proof. As $\langle s, D1 \rangle = 0$, Ds belongs to the range of L and $LL^+Ds = Ds$. The lemma then follows by applying Cauchy-Schwartz to the vectors $L^{1/2}x$ and $(L^+)^{-1/2}Ds$. \square

However, by taking $x = L^+Ds$, it is easy to verify that the lower bound is achieved with equality, so that L^+Ds must be an optimal solution for PROG'. This shows that, in order to compute an optimal solution, it suffices to solve the linear system of equations $Lx = Ds$. To complete the proof of Lemma 5.2, we notice that we can find an approximate solution of value $(1 + \varepsilon) \cdot \lambda_T$ in time $\tilde{O}(m \log(1/\varepsilon))$, by using the Spielman-Teng solver [ST03].

Interpretation as electrical flows. Here we briefly provide an interpretation of SPIMPROVE for $\beta = 1$ based on electrical flows and draw an analogy with the algorithm [AL08].

As shown in the previous section, the optimal vector v of the program PROG' satisfies $Lv = Ds$. As a scaled version of v is equivalent for the purposes of our rounding we can consider the following system of linear equations instead

$$Lx = D \left(\frac{1_T}{\text{vol}(T)} - \frac{1_{\bar{T}}}{\text{vol}(\bar{T})} \right).$$

If we then consider an electrical network where a vertex $i \in T$ has $\frac{d_i}{\text{vol}(T)}$ units of current fed into it and a vertex $i \in \bar{T}$ has $\frac{d_i}{\text{vol}(\bar{T})}$ extracted from it, we see that the resulting voltages must exactly obey the system of linear equations above. Hence, our optimal vector x can be interpreted as this setting of voltages. We believe that this interpretation will allow us to deduce more properties of the SPIMPROVE algorithm.

Electrical flows vs. graph flows. Finally, notice now the strong analogy between the flow-based algorithm of [AL08] and SPIMPROVE. In [AL08], each vertex i in T is input $\frac{d_i}{\text{vol}(T)}$ units of flow and each vertex i in \bar{T} outputs $\frac{d_i}{\text{vol}(\bar{T})}$ units of flow. Hence, the setup of the two algorithms is essentially the same, except for the use of electrical flows in place of standard graph flows.

6 Open Problems and Extensions

We hope that SPIMPROVE will be used as a black box in constructing new algorithms for graph partitioning in the same fashion as the flow-based cut-improvement algorithms have been. Moreover, we believe that the connection to electrical networks will open new avenues of investigation. Finally, an empirical evaluation of SPIMPROVE would provide an interesting comparison to the empirical results presented in [AL08] and [LMO09] about the flow-based cut-improvement method.

References

- [AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *STOC*, pages 227–236, 2007.
- [AL08] Reid Andersen and Kevin J. Lang. An algorithm for improving graph partitions. In *SODA*, pages 651–660, 2008.
- [AM85] Noga Alon and V. D. Milman. λ_1 , isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory, Ser. B*, 38(1):73–88, 1985.
- [ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 222–231, New York, NY, USA, 2004. ACM.
- [Chu97] Fan R.K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1997.
- [GGT89] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989.
- [GR98] Andrew V. Goldberg and Satish Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, 1998.
- [GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*, chapter The symmetric eigenvalue problem. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [KRV06] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 385–390, New York, NY, USA, 2006. ACM.
- [KVV00] R. Kannan, S. Vempala, and A. Veta. On clusterings-good, bad and spectral. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 367, Washington, DC, USA, 2000. IEEE Computer Society.
- [LLDM08] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR*, abs/0810.1355, 2008.
- [LMO09] Kevin J. Lang, Michael W. Mahoney, and Lorenzo Orecchia. Empirical evaluation of graph partitioning using spectral embeddings and flow. In *SEA*, pages 197–208, 2009.
- [LR04] Kevin Lang and Satish Rao. A flow-based method for improving the expansion or conductance of graph cuts. In *IPCO*, pages 325–337, 2004.
- [OSVV08] Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In *STOC*, pages 461–470, 2008.
- [OV09] Lorenzo Orecchia and Nisheeth K. Vishnoi. Towards a sdp-based approach to spectral methods: A nearly-linear time algorithm for graph partitioning. In *Manuscript*, 2009.
- [She09] Jonah Sherman. Breaking the multicommodity flow barrier for $\sqrt{\log n}$ -approximations to sparsest cut. In *FOCS '09: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009. To appear.

- [Shm97] David B. Shmoys. Cut problems and their application to divide-and-conquer. In *Approximation algorithms for NP-hard problems*, pages 192–235. PWS Publishing Co., Boston, MA, USA, 1997.
- [ST03] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. *CoRR*, cs.DS/0310051, 2003.

A Basic Facts

The following facts about K hold. The simple proofs are omitted.

Fact A.1. *Let $A, B \subseteq V$. Then*

1. $K(A, B) = K(\bar{A}, B)$,
2. $K(A, B) = K(A, \bar{B})$,
3. $K(A, B) \in [0, 1]$,
4. $K(A, B) = K(B, A)$,
5. $K(A, B) = 1$ if and only if $A = B$ or $\bar{A} = B$.