SIAM J. COMPUT.
Vol. 41, No. 6, pp. 1769–1786

MAX CUT AND THE SMALLEST EIGENVALUE[*]

LUCA TREVISAN[†]

Abstract. We describe a new approximation algorithm for Max Cut. Our algorithm runs in $\tilde{O}(n^2)$ time, where n is the number of vertices, and achieves an approximation ratio of .531. In instances in which an optimal solution cuts a $1-\varepsilon$ fraction of edges, our algorithm finds a solution that cuts a $1-4\sqrt{\varepsilon}+8\varepsilon-o(1)$ fraction of edges. Our main result is a variant of spectral partitioning, which can be implemented in nearly linear time. Given a graph in which the Max Cut optimum is a $1-\varepsilon$ fraction of edges, our spectral partitioning algorithm finds a set S of vertices and a bipartition $L, R = S - L$ of S such that at least a $1 - O(\sqrt{\varepsilon})$ fraction of the edges incident on S have one endpoint in L and one endpoint in R. (This can be seen as an analogue of Cheeger's inequality for the smallest eigenvalue of the adjacency matrix of a graph.) Iterating this procedure yields the approximation results stated above. A different, more complicated, variant of spectral partitioning leads to a polynomial time algorithm that cuts a $1/2 + e^{-\Omega(1/\varepsilon)}$ fraction of edges in graphs in which the optimum is $1/2 + \varepsilon$.

Key words. spectral graph theory, maximum cut

AMS subject classifications. 68W25, 05C50

DOI. 10.1137/090773714

1. Introduction.

In the Max Cut problem, we are given an undirected graph with nonnegative weights on the edges, and we wish to find a partition of the vertices (a *cut*) which maximizes the weight of edges whose endpoints are on different sides of the partition (such edges are said to be *cut* by the partition). We refer to the *cost* of a solution as the ratio between the total weight of cut edges and the total weight of all edges.

It is easy, given any graph, to find a solution that cuts half of the edges, providing an approximation factor of 1/2 for the problem. For example, a random partition cuts, on average, half of the total weight of edges, and a partition that cuts at least half of the total weight of edges can be found deterministically in linear time via a simple greedy procedure.

The algorithm of Goemans and Williamson [GW95], based on a semidefinite programming (SDP) relaxation, has a performance ratio of .878 . . . on general graphs, and it finds a cut of cost roughly $1 - \frac{2}{\pi}\sqrt{\varepsilon}$ in graphs in which the optimum is $1-\varepsilon$. Assuming the unique games conjecture, both results are the best possible for polynomial time algorithms [Kho02, KKMO04, MOO10] (see also [OW08]). A different rounding algorithm for the Goemans–Williamson relaxation, due to Charikar and Wirth [CW04], finds a solution of cost least a $1/2 + \Omega(\varepsilon/\log 1/\varepsilon)$ in graphs in which the optimum is $1/2+\varepsilon$. This result too is tight, assuming the unique games conjecture [KO06].

Arora and Kale [AK07] show that the Goemans–Williamson SDP relaxation can

[*] Received by the editors October 13, 2009; accepted for publication (in revised form) September 24, 2012; published electronically December 18, 2012. This material is based upon work supported by the National Science Foundation under grant CCF-0729137 and by the BSF under grant 2002246.
http://www.siam.org/journals/sicomp/41-6/77371.html

[†] Computer Science Division, U.C. Berkeley, Berkeley, CA 94720. Current address: Computer Science Department, Stanford University, Stanford, CA 94305-9025 (trevisan@stanford.edu).

be near-optimally solved in nearly linear time.[1]

No method other than SDP is known to yield an approximation better than 1/2 for Max Cut, and such approximation has been ruled out for large classes of linear programming relaxations [dlVKM07, STT07].

A main source of difficulty in designing approximation algorithms for Max Cut is the lack of good *upper bound* techniques for the Max Cut optimum of general graphs. Indeed, suppose that one is able to design and analyze a new polynomial time algorithm for Max Cut achieving, say, a .51 approximation ratio, and consider the behavior of the algorithm when given a graph whose Max Cut optimum is .501. Then the algorithm will clearly output a cut of cost $\leq .501$, but then the computations performed by the algorithm, plus the proof of its approximation ratio, provide a *certificate* that the optimum cut in the given graph is $\leq .501/.51 < .983$. The problem is that, except for SDP, we know of no technique that can provide, for every graph of Max Cut optimum $\leq .501$, a certificate that its optimum is $\leq .99$. Indeed, the results of [dlVKM07, STT07] show that large classes of linear programing relaxations of Max Cut are unable to distinguish instances in which the optimum is $\leq .501$ from instances in which the optimum is $\geq .99$.

One way around this obstruction would be to develop a new approximation algorithm that uses SDP only in the analysis, by showing that if the algorithm outputs a cut of cost c, then there is a dual solution for the Goemans–Williamson SDP relaxation of cost at most $c/.51$, thus proving that the Max Cut optimum is at most $c/.51$ and that the algorithm has a performance ratio at least .51. Such a primal-dual approach, which use a relaxation only in the analysis, has been applied to several problems based on *linear programming* relaxations, but unfortunately, as discussed above, linear programming relaxations are unlikely to be helpful in Max Cut approximation. As far as we know, the only examples of primal-dual approximation algorithms for combinatorial problems based on SDP are the algorithms for the sparsest cut problem described in [ARV09, KRV06, OSVV08].

Our results. Our main result is a variant of the spectral partitioning algorithm with the following property: given a graph $G = (V, E)$ in which the Max Cut optimum cost is $1 - \varepsilon$, it finds a set S and a partition of S into two disjoint sets of vertices L, R such that the weight of edges with one endpoint in L and one endpoint in R is at least a $1 - O(\sqrt{\varepsilon})$ fraction of the total weight of edges incident[2] on S. More precisely, we show that the weight of edges having both endpoints in L or both endpoints in R, plus half the weight of edges having an endpoint in S and an endpoint in $V - S$, is at most a $2\sqrt{\varepsilon} + o(1)$ fraction of the weight of edges incident on S. (See Theorem 1 and the subsequent discussion.) We will ignore the $o(1)$ additive factors in the rest of this section.

To derive an approximation algorithm for Max Cut, given a graph, we apply the partitioning algorithm and find sets L, R as above, we remove the vertices in $L \cup R$ from the graph, we recursively find a partition of the residual graph, and then we put back the vertices of L on one side of the partition and the vertices of R on the other side. We choose greedily the sides of the partition in which to put the vertices of L and R, so that edges of total weight at least half of the weight of edges between $L \cup R$ and $V - (L \cup R)$ are cut. This means that we cut all the edges that are cut in the

[1] The nearly linear time algorithm of Arora and Kale [AK07] applies only to graphs in which there is a bounded ratio between the largest degree and the smallest nonzero degree. In the appendix of [Tre08] we show that the same running time can be achieved on arbitrary graphs.

[2] An edge (i, j) is *incident* on a set S of vertices if at least one of the endpoints i, j belongs to S.

recursive step, plus all the edges with one endpoint in L and one endpoint in R, plus at least half of the weight edges between S and $V-S$. The recursion is stopped when less than half of the weight edges incident on S are cut, in which case we return a greedy partition of the residual graph.

We present an analysis of the recursive procedure due to Charikar, which improves an analysis of ours which appeared in a previous version of this paper. The basic idea is that if we look at a generic step of the execution of the algorithm, if the optimal solution in the original graph is $1-\varepsilon$, and the current residual graph holds a ρ fraction of the original edges, then we know that the optimum in the current residual graph is at least $1-\varepsilon/\rho$, and the spectral algorithm cuts at least a $1-2\sqrt{\varepsilon/\rho}$ fraction of the edges incident on $L \cup R$. When the recursion ends, it is because the spectral algorithm cuts less than half of the edges incident on $L \cup R$, and so the optimum of the residual graph at the end of the recursion must be less than 15/16, meaning that the residual graph at the end of the recursion contains at most a 16ε fraction of the edges of the original graph. Putting together this information, a calculation shows that the algorithm cuts at least a $1-4\sqrt{\varepsilon}+8\varepsilon$ fraction of edges of the graph. The ratio $(1-4\sqrt{\varepsilon}+8\varepsilon)/(1-\varepsilon)$ is always at least .531.

When applied to graphs in which the optimum is close to 1/2 (in fact, to any graph in which the optimum is smaller than 15/16), our algorithm may simply return a random partition. Thus, it fails to provide any nontrivial approximation to the Max Cut Gain problem, which is the same as the Max Cut problem, except that we count the number of cut edges *minus* $|E|/2$. (Equivalently, we count the number of cut edges minus the number of uncut edges.) For Max Cut Gain we develop a more sophisticated spectral partitioning algorithm with the following property: given a graph in which the Max Cut optimum is $1/2+\varepsilon$, our algorithm finds sets L, R such that the number of edges incident on $L \cup R$ cut by the partition exceeds the number of uncut edges by at least a $1/\exp(\Omega(1/\varepsilon))$ fraction of the edges incident on $L \cup R$. Iterating this algorithm allows us to find a cut for the entire graph of cost at least $1/2+1/\exp(\Omega(1/\varepsilon))$.

This second algorithm can also be applied to the case in which edges have negative weights, and it approximates a general class of quadratic programs. Given a symmetric real-valued matrix Q with zeros on the diagonal, if there exists a vector $x \in \{-1,1\}^V$ such that $x^TQx \geq \varepsilon \cdot ||Q||_1$, our algorithm finds a vector $y \in \{-1,1\}^V$ such that $y^TQy \geq \exp(-O(1/\varepsilon)) \cdot ||Q||_1$, where $||Q||_1 := \sum_{i,j} |Q(i,j)|$. (The algorithm of Charikar and Wirth finds a vector y such that $y^TQy \geq ||Q||_1 \cdot \varepsilon/\log 1/\varepsilon$.)

Relation to Cheeger's inequality. In the case of regular graphs, our main result, Theorem 1, may be seen as an analogue of Cheeger's inequality [Alo86] for the smallest (rather than second largest) eigenvalue of the adjacency matrix of the graph. We discuss this analogy in section 5.

Relation to the Goemans–Williamson relaxation. Our algorithm may also be seen as a primal-dual algorithm that produces, along with a cut, a feasible solution to the semidefinite dual of the Goemans–Williamson relaxation such that the cost of the cut is at least .531 times the cost of the dual solution. We describe this view in section 6.

Other relations to previous work. It was known that one can use spectral methods to certify an upper bound to the Max Cut optimum of a given graph. In particular, if G is a d-regular graph of adjacency matrix A, and $M := \frac{1}{d}A$ has eigenvalues

$1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, then one can easily show[3] that

$$\text{Max Cut} \leq \frac{1}{2} + \frac{1}{2}|\lambda_n|. \tag{1}$$

(Our Lemma 2 is essentially a restatement of this fact.)

What is new is that we are able to prove a *converse*, in Lemma 3, and show that a nontrivial consequence follows whenever $|\lambda_n|$ is close to 1.

As mentioned above, it was known that $\lambda_n = -1$ if and only if G has a bipartite connected component. In particular, if G is connected and not bipartite, then $\lambda_n > -1$. Alon and Sudakov [AS00] consider the question of how small, in such a case, the gap $1 - |\lambda_n|$ can be. They show that, if G is connected and not bipartite, it has maximum degree d and diameter D, and λ_n is the smallest eigenvalue of the adjacency matrix A, then $d - |\lambda_n| \geq \frac{1}{(D+1)\cdot n}$. This was improved to $d - |\lambda_n| \geq \frac{1}{D \cdot n}$ by Nikiforov [Nik07]. Our result implies the weaker bound $d - |\lambda_n| \geq \frac{1}{dn^2}$ in a d-regular graph.

The "converse expander mixing lemma" of Bilu and Linial [BL06] has some similarity with our approach to Max Cut Gain. Bilu and Linial show that if G is a d-regular graph, A is the adjacency matrix, and $\lambda_1 \geq \cdots \geq \lambda_n$ are the eigenvalues of $M := \frac{1}{d}A$, then if $\max\{\lambda_2, |\lambda_n|\} \geq \varepsilon$, it follows that there are sets L, R such that the number of edges between L and R differs from what one would expect in a random d-regular graph by a multiplicative error factor $\Omega(\varepsilon/\log 1/\varepsilon)$. In our main result for Max Cut Gain (Theorem 11) we have a stronger assumption, that $|\lambda_n| \geq \varepsilon$, but we need to derive a much stronger conclusion, namely, that the number of edges between L and R not only exceeds the number of edges that one would expect in a random d-regular graph but in fact exceeds the number of edges which are entirely contained in L or entirely contained in R.

The main difference between our proof and the proof of Bilu and Linial is that the combinatorial quantity that they relate to $\max\{\lambda_2, |\lambda_n|\}$ is the normalized multilinear form $\max_{x,y\in\{-1,0,1\}^V} |x^T M y|/(||x|| \cdot ||y||)$ for a certain matrix M, while the combinatorial quantity that we wish to relate to $|\lambda_n|$ is the normalized homogeneous quadratic form $\max_{x\in\{-1,0,1\}^V} |x^T M x|/||x||^2$ for a different matrix M. Generally, it is considerably harder to round continuous relaxations of quadratic forms of the latter type compared to multilinear forms of the first kind. (See, e.g., the introduction of [CW04] and their discussion of their results versus the results of Alon and Naor [AN06].)

The idea of iteratively removing parts of an instance in which one has a good solution appears in various works on the sparsest cut problem (for example, in the way Spielman and Teng [ST04] find a balanced separator using their "nibble" procedure), and it was used to approximate the Max Cut problem (in the version in which one wants to minimize the number of uncut vertices) by Agarwal et al. [ACMM05]. In the algorithm of Agarwal et al., as in our algorithm, the basic procedure that is being iterated finds a set S of vertices and a bipartition L, R of S such that most of the edges incident on S have one endpoint in L and one endpoint in R.

[3] Inequality (1) appears to be a folklore result. Lovász [Lov03, Proposition 4.4] credits it to Delorme and Poljak [DP93a, DP93b], and it also appears in [Bra98, Theorem 1.4]. The earliest related reference we are aware of is [Hae79, Theorem 2.1.4.i], which states that if V_1, V_2 is a partition of the vertices of a d-regular graph $G = (V, E)$, if d_1 is the average degree of the subgraph induced by V_1, and λ_n is the smallest eigenvalue of the adjacency matrix of G, then $n_1 d - n d_1 \leq -\lambda_n \cdot (n - n_1)$, from which one can derive that $n_1 \cdot (d - d_1)$, the number of edges crossing the cut, obeys $n_1 \cdot (d - d_1) \leq n_1 \cdot (n - n_1) \cdot (d - \lambda_n)/n$, and the latter term is at most $n \cdot (d - \lambda_n)/4$.

Subsequent work. Soto [Sot09] improves the analysis of our Max Cut algorithm and proves that it achieves an approximation ratio of .614. It remains open to provide a tight analysis of the algorithm.

Our algorithm for Max Cut Gain provides only an exponentially weak approximation guarantee: in a graph in which the Max Cut optimum is $\frac{1}{2}+\varepsilon$ we show how to find a cut of cost $\frac{1}{2}+\exp(-\varepsilon^{-O(1)})$. The exponential loss comes from our analysis of the relation between $|\lambda_n|$ and a combinatorial parameter that we call the *gain ratio* of the graph. Bhaskara, Charikar, and Vijayaraghavan [BCV09] construct a graph with an exponential gap between $|\lambda_n|$ and the gain ratio, showing that our analysis cannot be significantly improved.

2. Sparsification.

It follows from the Chernoff bound that if we are given a graph $G=(V,E)$ and we sample $O(\delta^{-2}|V|)$ edges with replacement,[4] then, with high probability, every cut $(S,\bar{S})$ has the same cost in the original graph as in the new graph, up to an additive error δ.[5] Recalling that the optimum of the Max Cut problem is always at least $1/2$, an arbitrarily small additive approximation error translates to an arbitrarily small multiplicative error.

For this reason, all the dependency on $|E|$ in the running time of our algorithm can be changed to a dependency on $|V|$ with an arbitrarily small loss in the approximation factor.

3. The spectral algorithm.

In this section we prove our main result.

THEOREM 1 (main). *There is an algorithm that, given a graph $G=(V,E)$ for which the optimum of the Max Cut problem is at least $1-\varepsilon$, and a parameter δ, finds a vector $y\in\{-1,0,1\}^V$ such that*

$$\frac{\sum_{i,j} A_{i,j}|y_i+y_j|}{\sum_i d_i|y_i|} \le 4\sqrt{\varepsilon}+\delta,$$

where $A_{i,j}$ is the weight of edge (i,j) and d_i is the (weighted) degree of vertex i.

The algorithm can be implemented in nearly linear randomized time $O(\delta^{-2}\cdot(|V|+|E|)\cdot\log|V|)$.

To understand the statement of Theorem 1, let y be the vector returned by the algorithm, and call L the set of vertices with negative coordinates in y and R the set of vertices with positive coordinates. Then, up to constant factors, the numerator counts the number of edges incident on $L\cup R$ which fail to have one endpoint in L and one endpoint in R, and the denominator counts the number of edges incident on S. More specifically, the numerator counts four times the edges that are entirely contained in L or entirely contained in R and twice the edges that have one endpoint in S and one endpoint in $V-S$. The denominator counts every edge incident on $L\cup R$ once or twice, depending on whether one or both of the endpoints of the edge are in S.

The following form of the conclusion of Theorem 1 will be convenient in our analysis: given the vector y, call M the weight of edges incident on $L\cup R$, U the weight of "uncut" edges that have both endpoints in L or both endpoints in R, and

[4] If the graph is unweighted, we sample from the uniform distribution over the edges; otherwise we sample from the distribution in which each edge has a probability proportional to its weight.

[5] Note that the sparsified graph is an unweighted multigraph and that the sparsification process is considerably simpler than that used for algorithms for sparsest cut and other graph minimization problems.

X the weight of "cross" edges that have exactly one endpoint in $L \cup R$; then

$$U + \frac{1}{2} X \leq \left(2\sqrt{\varepsilon} + \frac{\delta}{2} \right) \cdot M.$$

Let A be the adjacency matrix of our input graph G (hence $A_{i,j}$ is the weight of the edge between i and j), and let D be the diagonal matrix such that $D_{i,i}$ is the weighted degree d_i of vertex i and $D_{i,j} = 0$ for $i \neq j$.

Theorem 1 follows by combining the following two results and noting that, for $a, b \geq 0$, $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.

Lemma 2. *If the optimum Max Cut in G has cost at least $1 - \varepsilon$, there is a vector $x \in \mathbb{R}^V$ such that*

$$x^T (D + A) x \leq 2\varepsilon \cdot x^T D x.$$

Furthermore, for every $\delta > 0$, we can find in randomized time $O(\delta^{-1} \cdot (|E| + |V|) \cdot \log |V|)$ a vector $x \in \mathbb{R}^V$ such that

$$x^T (D + A) x \leq (2\varepsilon + \delta) \cdot x^T D x.$$

Lemma 3. *Given a vector $x \in \mathbb{R}^V$ such that $x^T (D + A) x \leq \varepsilon \cdot x^T D x$, we can find in time $O(|E| + |V| \log |V|)$ a vector $y \in \{-1, 0, 1\}^V$ such that*

$$\frac{\sum_{i,j} A_{i,j} |y_i + y_j|}{\sum_i d_i |y_i|} \leq \sqrt{8\varepsilon}. \tag{2}$$

Lemma 2 has a simple proof, and it can be seen as a statement about the semidefinite dual of the Goemans–Williamson relaxation, as discussed in section 6. Lemma 3 is the main result of this paper.

3.1. Proof of Lemma 2.

Consider the optimization problem

$$\min_{x \in \mathbb{R}^V} \frac{x^T A x}{x^T D x}. \tag{3}$$

Let $(S, \bar{S})$ be an optimum cut for G, and define the vector $x^* \in \{-1, 1\}^V$ such that $x^*_i = 1$ if $i \in S$ and $x^*_i = -1$ otherwise. Then $x^{*T} A x^*$ equals twice the difference between the weight of edges not cut by $(S, \bar{S})$ and the weight of edges that are cut, which is at most $2 \cdot (2\varepsilon - 1) \cdot |E|$. As for $x^{*T} D x^*$, we have

$$x^{*T} D x^* = \sum_i d_i \cdot (x^*_i)^2 = \sum_i d_i = 2 \cdot |E|.$$

Thus x^* is a feasible solution to (3) of cost at most $2\varepsilon - 1$, and if $\hat{x}$ is the optimal solution to (3), then we must have

$$\hat{x}^T A \hat{x} \leq (2\varepsilon - 1) \hat{x}^T D \hat{x}.$$

To prove the "furthermore" part of the lemma, we observe that the optimization problem in (3) is equivalent to

$$\min_{x \in \mathbb{R}^V} \frac{x^T D^{-1/2} A D^{-1/2} x}{x^T x}, \tag{4}$$

where $D^{-1/2}$ is the matrix that such that $D^{-1/2}_{i,j} = 0$ if $D_{i,j} = 0$ and $D^{-1/2}_{i,j} = 1/\sqrt{D_{i,j}}$ otherwise. In turn, the optimization problem in (4) is the problem of computing the smallest eigenvalue of $D^{-1/2}AD^{-1/2}$, which is the same as computing the largest eigenvalue of the positive semidefinite matrix $I - D^{-1/2}AD^{-1/2}$.

Given an $n \times n$ positive semidefinite matrix M with T nonzero entries and of largest eigenvalue λ_1, and a parameter δ, it is possible to find a vector x such that $x^T Mx \geq \lambda_1 \cdot (1-\delta) \cdot x^T x$ in randomized time $O(\delta^{-1} \cdot (T+n) \cdot \log n)$ [KW92]. Applying the algorithm to $I - D^{-1/2}AD^{-1/2}$, which, as proved above, has a largest eigenvalue which is at least $2-2\varepsilon$, and which has $|E|+|V|$ nonzero entries, we find in randomized time $O(\delta^{-2} \cdot (|E| + |V|) \cdot \log |V|)$ a vector x' such that

$$\frac{x'^T (I - D^{-1/2}AD^{-1/2})x'}{x'^T x'} \geq 2 - 2\varepsilon - \delta,$$

and so

$$x'^T D^{-1/2}AD^{-1/2}x' \leq (2\varepsilon + \delta - 1) \cdot x'^T x',$$

and, if we define $x'' := x'D^{1/2}$, then

$$x''^T Ax'' \leq (2\varepsilon + \delta - 1)x''^T Dx'',$$

which we can rewrite as

$$x''^T (A + D)x'' \leq (2\varepsilon + \delta)x''^T Dx'' \ .$$

3.2. Proof of Lemma 3.

We now come to our main result.

The condition $x(D + A)x \leq \varepsilon \cdot xDx$ is equivalent to

$$\frac{1}{2}\sum_{i,j} A_{i,j}(x_i + x_j)^2 \leq \varepsilon \sum_i d_i x_i^2. \tag{5}$$

Before starting the formal proof, we describe a heuristic argument that gives some intuition for the actual proof.

Proof idea. Inequality (5) states that the average value of $(x_i + x_j)^2$, for an edge (i, j), is at most ε times the average value of x_i^2 and x_j^2. So, nonrigorously, we would guess that for a typical edge the value of $|x_i + x_j|$ is at most about $\sqrt{\varepsilon}$ times $|x_i| + |x_j|$. For this to happen, it must be the case that x_i and x_j have different signs, and their absolute value is nearly the same; that is, for some positive c, $x_i = -c$ and $x_j = c(1 - \sqrt{\varepsilon})$. Suppose now that we pick a random threshold t, and we define $y_i = -1 \Leftrightarrow x_i \leq -t$ and $y_i = 1 \Leftrightarrow x_i \geq t$. Then $|y_i - y_j|$ is 2 with probability $c\sqrt{\varepsilon}$ and zero otherwise, while $|y_i|$ and $|y_j|$ are 1 with probability roughly c and zero otherwise; then it follows that the expectation of $\sum_{(i,j)} |y_i + y_j|$ is about a $\sqrt{\varepsilon}$ fraction of the expectation of $\sum_i d_i|y_i|$.

Our algorithm, which we call the 2-Thresholds Spectral Cut algorithm and abbreviate 2TSC, is as follows.

Algorithm 2TSC.

```
• For every vertex $k$
    – Define the vector $y^k \in \{-1, 0, 1\}^V$ as follows:
          $y^k_i = -1$   iff   $x_i < -|x_k|$,
          $y^k_i = 1$    iff   $x_i > |x_k|$,
          $y^k_i = 0$    iff   $|x_i| \leq |x_k|$.
```

- Output the vector y^k for which the ratio

$$\frac{\sum_{i,j} A_{i,j}|y^k_i + y^k_j|}{\sum_i d_i|y^k_i|}$$

is smallest.

The algorithm can be implemented to run in $O(|E| + |V|\log|V|)$ time. We first sort the vertices according to the value of $|x_i|$, and so we assume we have $|x_1| \leq |x_2| \leq \cdots \leq |x_n|$ when we run 2TSC. At each step k, we need to modify the vector y only in positions k and $k-1$, and the cost of recomputing the ratio is only $O(d_k + d_{k-1})$, so that all the n steps together take time $O(|E|)$.

We need to argue that, under the assumption of the lemma, the algorithm outputs a vector y such that the ratio in (2) is at most $\sqrt{8\varepsilon}$.

In order to analyze 2TSC, we study the following randomized process:

- Pick a value t uniformly in $[0, \max_i x_i^2]$.
- Define $Y \in \{-1, 0, 1\}^V$ as follows:

$$\begin{array}{lll} Y_i = -1 & \text{iff} & x_i < -\sqrt{t}, \\ Y_i = 1 & \text{iff} & x_i > \sqrt{t}, \\ Y_i = 0 & \text{iff} & |x_i| \leq \sqrt{t}. \end{array}$$

Every Y that is generated by the probabilistic process with positive probability is considered by Algorithm 2TSC at some stage; this implies that if Algorithm 2TSC outputs a vector y such that $\sum_{i,j} A_{i,j}|y(i) + y(j)| > \sqrt{8\varepsilon}\sum_i d_i|y_i|$, then in the randomized process we must have $\sum_{i,j} A_{i,j}|Y(i)+Y(j)| > \sqrt{8\varepsilon}\sum_i d_i|Y_i|$ with probability 1 and, in particular, $\mathbb{E}\sum_{i,j} A_{i,j}|Y(i) + Y(j)| > \sqrt{8\varepsilon}\,\mathbb{E}\sum_i d_i|Y_i|$.

We shall prove that $\mathbb{E}\sum_{i,j} A_{i,j}|Y(i) + Y(j)| \leq \sqrt{8\varepsilon}\,\mathbb{E}\sum_i d_i|Y_i|$, and so we shall conclude that the output of Algorithm 2TSC satisfies the claim.

Since (5) and the distribution Y are invariant under multiplying x by a scalar, we may assume that $\max_i |x_i| = 1$, so that t is chosen uniformly in $[0, 1]$.

A case analysis shows that, for every edge (i, j),

$$\mathbb{E}\,|Y_i + Y_j| \leq |x_i + x_j| \cdot (|x_i| + |x_j|). \tag{6}$$

To verify (6) we need to distinguish the case in which x_i and x_j have different signs from the case in which they have the same sign. We assume without loss of generality that $|x_i| > |x_j|$.

- If they have different signs, and, say, $|x_i| > |x_j|$, then $|Y_i + Y_j| = 1$ when $|x_j|^2 \leq t \leq |x_i|^2$ and zero otherwise. Indeed, if $t < |x_j|^2$, then $Y_i = -Y_j$ and $|Y_i + Y_j| = 0$, and if $t > |x_i|^2$, then $Y_i = Y_j = 0$.
  So $\mathbb{E}\,|Y_i + Y_j|$ equals $|x_i|^2 - |x_j|^2$, which is equal to the right-hand side of (6).
- If they have the same sign, then $|Y_i + Y_j| = 2$ when $t \leq |x_j|^2$, $|Y_i + Y_j| = 1$ when $|x_j|^2 < t \leq |x_i|^2$, and $|Y_i + Y_j| = 0$ when $t > |x_i|^2$.
  Overall, $\mathbb{E}\,|Y_i + Y_j|$ equals $2x_j^2 + (x_i^2 - x_j^2) = x_j^2 + x_i^2$. The right-hand side of (6) is $(x_i + x_j)^2$, which is only larger.

Note also that $\mathbb{E}\,|Y_i| = x_i^2$.

To complete our argument it remains to apply Cauchy–Schwarz and standard manipulations:

$$\begin{aligned}\mathbb{E}\sum_{i,j} A_{i,j}|Y_i + Y_j| &\le \sum_{i,j} A_{i,j}|x_i + x_j| \cdot (|x_i| + |x_j|)\\ &\le \sqrt{\sum_{i,j} A_{i,j}|x_i + x_j|^2} \cdot \sqrt{\sum_{i,j} A_{i,j}(|x_i| + |x_j|)^2}\ .\end{aligned}$$

By our assumption,

$$\sum_{i,j} A_{i,j}|x_i + x_j|^2 \le 2\varepsilon \sum_i d_i x_i^2,$$

and it is a standard calculation that

$$\sum_{i,j} A_{i,j}(|x_i| + |x_j|)^2 \le 2\sum_{i,j} A_{i,j}(|x_i|^2 + |x_j|^2) = 4\sum_i d_i x_i^2,$$

and so

$$\mathbb{E}\sum_{i,j} A_{i,j}|Y_i + Y_j| \le \sqrt{8\varepsilon}\sum_i d_i x_i^2 = \sqrt{8\varepsilon}\,\mathbb{E}\sum_i d_i|Y_i|.$$

This completes the proof that Algorithm 2TSC performs as required by the lemma.

4. Approximation for Max Cut.

In this section we analyze the following algorithm.

Algorithm Recursive-Spectral-Cut.

- Input: graph $G = (V, E)$, accuracy parameter δ.
- Run the algorithm of Theorem 1 with accuracy parameter δ, and let $y \in \{-1, 0, 1\}$ be the solution found by the algorithm; call M the weighted number of edges (i, j) such that least one of y_i or y_j is nonzero, C is the weighted number of *cut* edges (i, j) such that y_i, y_j are both nonzero and have opposite signs, and X is the weighted number of *cross* edges (i, j) such that exactly one of y_i, y_j is zero.
- If $C + \frac{1}{2}X \le \frac{1}{2}M$, then use a greedy algorithm to find a partition of V that cuts $\ge |E|/2$ edges, and return it.
- If $C + \frac{1}{2}X > \frac{1}{2}M$, then let $L := \{i : y_i = -1\}$, $R := \{i : y_i = 1\}$, and $V' := \{i : y_i = 0\}$, let $G' = (V', E')$ be the graph induced by V', recursively call Recursive-Spectral-Cut on G', and let V_1, V_2 be the partition found by the algorithm; return $(V_1 \cup L, V_2 \cup R)$ or $(V_1 \cup R, V_2 \cup L)$—whichever is better.

Note that the algorithm runs in randomized time $O(\delta^{-2} \cdot |V| \cdot (|V| + |E|) \cdot \log |V|)$ because each recursive call takes time $O(\delta^{-2} \cdot (|V| + |E|) \cdot \log |V|)$ and there are at most $|V|$ calls.

In a preliminary version of this paper we presented a simple argument showing that if *opt* $\ge 1 - \varepsilon$, then the algorithm cuts at least a $1 - O(\varepsilon^{1/3}) - \delta$ fraction of edges. The following tighter argument is due to Moses Charikar (personal communication, July 2008).

Theorem 4. *If Algorithm* Recursive-Spectral-Cut *receives in input a graph $G = (V, E)$ whose optimum is $1 - \varepsilon$, with $\varepsilon < 1/16$, then it finds a solution that cuts at least a $1 - 4\sqrt{\varepsilon} + 8\varepsilon - \frac{\delta}{2}$ fraction of edges.*

Proof. Consider the tth recursive call of the algorithm, and let G_t be the residual graph at that point, and let $\rho_t \cdot |E|$ be the number of edges of G_t. Then we observe that the Max Cut optimum in G_t is at least $1 - \varepsilon/\rho_t$.

Let S_t be the set of vertices and L_t, R_t the partition found by the algorithm of Theorem 1. Let G_{t+1} be the residual graph at the following step and $\rho_{t+1} \cdot |E|$ the number of edges of G_{t+1}. (If the algorithm stops at the tth call, we shall take G_{t+1} to be the empty graph; if the algorithm discards L_t, R_t and chooses a greedy cut, we shall take G_{t+1} to be empty and L_t, R_t to be the partition given by the greedy cut.)

We know by Theorem 1 that the algorithm will cut at least a $1 - 2\sqrt{\varepsilon/\rho_t} - \delta/2$ fraction of the $|E| \cdot (\rho_t - \rho_{t+1})$ edges incident on S_t.

Indeed, we know that at least a $\max\{1/2, 1 - 2\sqrt{\varepsilon/\rho_t} - \delta/2\}$ fraction of those edges are cut (for small value of ρ_t, it is possible that $1 - 2\sqrt{\varepsilon/\rho_t} + \delta/2 < 1/2$, but the algorithm is always guaranteed to cut at least half of the edges incident on S_t). This means that any convex combination of $1/2$ and $1 - 2\sqrt{\varepsilon/\rho_t} - \delta/2$ is still a lower bound on the fraction of edges incident on S_t cut by the algorithm.

If both ρ_t and ρ_{t+1} are at least 16ε, we are going to use the lower bound

$$\begin{aligned}|E| \cdot (\rho_t - \rho_{t+1}) \cdot \left(1 - 2\sqrt{\frac{\varepsilon}{\rho_t}} - \frac{\delta}{2}\right) &= |E| \int_{\rho_{t+1}}^{\rho_t} \left(1 - 2\sqrt{\frac{\varepsilon}{\rho_t}} - \frac{\delta}{2}\right) dr \\ &\geq |E| \int_{\rho_{t+1}}^{\rho_t} \left(1 - 2\sqrt{\frac{\varepsilon}{r}} + \frac{\delta}{2}\right) dr.\end{aligned}$$

If $\rho_t \geq 16\varepsilon \geq \rho_{t+1}$, then we use the lower bound

$$\begin{aligned}&|E| \cdot (\rho_t - 16\varepsilon) \cdot \left(1 - 2\sqrt{\frac{\varepsilon}{\rho_t}} + \frac{\delta}{2}\right) + |E| \cdot (16\varepsilon - \rho_{t+1}) \cdot \frac{1}{2} \\ &\geq |E| \int_{16\varepsilon}^{\rho_t} \left(1 - 2\sqrt{\frac{\varepsilon}{r}} - \frac{\delta}{2}\right) dr + |E| \cdot \int_{\rho_{t+1}}^{16\varepsilon} \frac{1}{2} dr.\end{aligned}$$

Finally, if both ρ_t and ρ_{t+1} are smaller than 16ε, we use the lower bound

$$|E| \cdot (\rho_t - \rho_{t+1}) \cdot \frac{1}{2} = |E| \cdot \int_{\rho_{t+1}}^{\rho_t} \frac{1}{2} dr.$$

Summing those bounds, we have that the number of edges cut by the algorithm is at least

$$\begin{aligned}&|E| \cdot \left(\int_{16\varepsilon}^{1} \left(1 - 2\sqrt{\frac{\varepsilon}{r}} - \frac{\delta}{2}\right) dr + \int_{0}^{16\varepsilon} \frac{1}{2} dr\right) \\ &= |E| \cdot \left(1 - 4\sqrt{\varepsilon} + 8\varepsilon - (1 - 16\varepsilon)\frac{\delta}{2}\right). \qquad \square\end{aligned}$$

COROLLARY 5. *Algorithm* RECURSIVE-SPECTRAL-CUT *is a* $.531128 - \delta$ *approximate algorithm for Max Cut.*

Proof. Write $opt = 1 - \varepsilon$. If $\varepsilon > 1/16$, then the algorithm finds a solution of cost $> 1/2$, and the approximation ratio is $16/30 > 5.33333$.

If $0 \leq \varepsilon \leq 1/16$, then the algorithm finds a solution of cost at least $1 - 4\sqrt{\varepsilon} + 8\varepsilon - \delta/2$, and the approximation ratio is at least

$$\frac{1 - 4\sqrt{\varepsilon} + 8\varepsilon - \delta/2}{1 - \varepsilon} \geq \frac{1 - 4\sqrt{\varepsilon} + 8\varepsilon}{1 - \varepsilon} - \delta.$$

If we call $\rho(\varepsilon) := \frac{1-4\sqrt{\varepsilon}+8\varepsilon}{1-\varepsilon}$, then some calculus shows that, for $0 \leq \varepsilon \leq 1/16$, $\rho(\varepsilon)$ is minimized at $\varepsilon = \frac{73-9\sqrt{65}}{8} \approx .05496$ and is always at least $\frac{1}{2}(\sqrt{65}-7) > .531128$. □

5. Relation to Cheeger's inequality.

In this section we compare our main result, Theorem 1, with Cheeger's inequality [Alo86].

If G is an undirected graph, A is its adjacency matrix, D is the diagonal matrix such that $D(v,v) = d(v)$, where $d(v)$ is the degree $\sum_z A(v,z)$ of vertex v, and M is the normalized adjacency matrix $M := D^{-1/2}AD^{-1/2}$, then M has n eigenvalues, counting multiplicities, which we shall call $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. It is always the case that $\lambda_1 = 1$, and that $|\lambda_i| \leq 1$ for every i. The extremal cases are captured by the following well-known facts:

1. $\lambda_2 = 1$ if and only if G is disconnected, that is, if and only if there is a set S, $|S| \leq |V|/2$, such that no edge of G leaves S.
2. $\lambda_n = -1$ if and only if G contains a bipartite connected component, that is, if and only if there is a set S and a partition of S into disjoint sets L, R, such that all edges incident on S have one endpoint in L and one endpoint in R.

Cheeger's inequality characterizes the cases in which λ_2 is close to 1 as those in which there is a set S, $|S| \leq |V|/2$, such that the number of edges between S and $V - S$ is small compared to $d|S|$.

If we define $h(G)$ to be the *conductance* of G,

$$h(G) = \min_{S\subseteq V:\ |S|\leq|V|/2} \frac{edges(S, V-S)}{\sum_{v\in S} d(v)},$$

then we have Cheeger's inequality

$$\sqrt{2\cdot(1-\lambda_2)} \geq h(g) \geq \frac{1}{2}\cdot(1-\lambda_2). \tag{7}$$

Similarly, Lemmas 2 and 3 characterize the cases in which λ_n is close to -1 as those in which there is a set S and a partition (L,R) of S such that the number of edges incident on S which fail to be cut by the partition is small compared to $d|S|$.

Define the *bipartiteness ratio* number of a graph to be

$$\beta(G) := \min_{y\in\{-1,0,1\}^V} \frac{\sum_{i,j}|y_i + y_j|}{2d\sum_i |y_i|},$$

which is equivalent to

$$\beta(G) = \min_{S\subseteq V,\ (L,R)\text{ partition of }S} \frac{2edges(L) + 2edges(R) + edges(S, V-S)}{\sum_{v\in S} d(v)};$$

then we have

$$\sqrt{2\cdot(1-|\lambda_n|)} \geq \beta(G) \geq \frac{1}{2}\cdot(1-|\lambda_n|). \tag{8}$$

There are examples in which both inequalities in (8) are tight within constant factors.

If we take an odd cycle with n vertices, then $\beta(G) \geq \frac{1}{n}$, because for every subset S of vertices and for every bipartition of S there is at least one failed edge, and the number of edges incident on S is at most n. In an odd cycle, however, $d(v) = 2$ for every vertex and $|\lambda_n| = 2 - O(1/n^2)$, and so β is as large as $\Omega(\sqrt{1 - |\lambda_n|})$.

To see the tightness of the other inequality, start from a k-regular expander such that, say, $\max\{\lambda_2, |\lambda_n|\} \leq 1/2$. (Such graphs exist for constant k.) Then construct G by taking the disjoint union of the edges of G and the edges of a $k \cdot (1-\varepsilon)/\varepsilon$-regular bipartite graph, so that the resulting graph is d-regular with $d := k/\varepsilon$. There is a cut that cuts all the edges of the bipartite graph, so $\beta(G) \leq \varepsilon$, but the smallest eigenvalue of M is at least $-1 + k/2d \geq -1 + \varepsilon/2$, meaning that β is $O(1 - |\lambda_n(G)|)$.

Our results, as stated in (8), are not just syntactically similar to Cheeger's inequality; there are also similarities between the proof of Cheeger's inequality and that of Theorem 1. The analysis in Cheeger's inequality relies on the study of the quadratic form

$$\sum_{i,j} A(i,j) \cdot (x_i - x_j)^2, \tag{9}$$

and it is based on the intuition that if (9) is small compared to $\sum_i x_i^2$, then for most edges (i,j) we have $x_i \approx x_j$.

Our analysis was based on the study of the quadratic form

$$\sum_{i,j} A(i,j) \cdot (x_i + x_j)^2 \tag{10}$$

and the intuition that if (10) is small compared to $\sum_i x_i^2$, then for most edges we have $x_i \approx -x_j$.

6. Relation to the Goemans–Williamson relaxation.

The dual of the Goemans–Williamson relaxation is

$$\begin{array}{l} \min |E| - \frac{1}{4}\sum_i y_i \\ \text{subject to} \\ D + A - \operatorname{diag}(y_1, \dots, y_n) \succeq 0. \end{array} \tag{11}$$

We can see Lemma 2 as stating a special case of the weak duality fact that the cost of every feasible solution to (11) is an upper bound to the optimal cut in the graph.

Indeed, if the optimal cut is of size $> |E| \cdot (1-\varepsilon)$, then no solution of cost $\leq |E| \cdot (1-\varepsilon)$ can be feasible for (11). In particular, the solution $y_i = 2\varepsilon d_i$ has cost $1 - \varepsilon$ and cannot be feasible, meaning that $D(1-2\varepsilon) + A$ cannot be feasible, and there is a vector x such that $x(D(1-2\varepsilon) + A)x < 0$.

In turn, Lemma 3 has the following primal-dual interpretation: given a graph G, there is an ε such that Algorithm 2TSC finds L, R such that $C + \frac{1}{2}X \geq (1 - 2\sqrt{\varepsilon} - \delta/2)M$, and the solution $y_i := 2\varepsilon d_i$ is feasible for (11), thus showing that the Max Cut optimum is at most $1 - \varepsilon$.

Given this premise, we can now view Algorithm Recursive-Spectral-Cut as a primal-dual algorithm.

At step t of the recursion, let $\rho_t|E|$ be the number of edges in the residual graph G_t, and let C_t and X_t be the numbers of cut and cross edges in the solution L_t, R_t found by the algorithm. Define ε_t so that $1 - \varepsilon_t/\rho_t$ is the upper bound on the Max Cut of G_t given by the dual solution associated to the algorithm as above, and the algorithm satisfies $C_t + \frac{1}{2}X_t \geq (1 - 2\sqrt{\varepsilon_t/\rho_t} - \delta/2)M_t$. Then the dual solution at time t also proves an upper bound $1 - \varepsilon_t$ to the Max Cut optimum of G. Let $\varepsilon := \max_t \varepsilon_t$;

then we have (i) a dual solution proving that the Max Cut of G is $\leq 1-\varepsilon$, and we know that (ii) at every step t we have $C_t + \frac{1}{2}X_t \geq (1 - 2\sqrt{\varepsilon/\rho_t} - \delta/2)M_t$. From fact (ii) and the analysis done in the proof of Theorem 1 we see that the algorithm outputs a solution that cuts at least a $1 - 4\sqrt{\varepsilon} + 8\varepsilon - \delta/2$ fraction of edges, and it is able to output a feasible dual solution to the Goemans–Williamson relaxation proving a $1-\varepsilon$ upper bound to the optimum.

In particular, the ratio between the cost of the solution found by the algorithm and the upper bound provided by the dual solution is always at least .531.

7. Quadratic programming and the Max Cut Gain problem.

Let A be the adjacency matrix of a weighted graph with no self-loops, possibly with negative weights, let $d_i := \sum_j |A_{i,j}|$ be the weighted degree of node i, and let $D := \operatorname{diag}(d_1, \ldots, d_n)$. *Max Cut Gain* is the optimization problem

$$\max_{y \in \{-1,1\}^V} -\frac{y^T Ay}{y^T Dy}. \tag{12}$$

In words, Max Cut Gain is the maximum, over all cuts, of the difference between the number of cut edges and the number of edges that are not cut, divided by the total number of edges. Equivalently, the optimum of Max Cut Gain is ε if and only if the optimum of Max Cut is $\frac{1}{2} + \frac{1}{2}\varepsilon$. (The name of the problem comes from the fact that one is measuring how much one *gains* by using an optimum cut compared to a random cut, which cuts only a 1/2 fraction of edges.)

Note that, up to the scaling that we do by dividing by $y^T Dy = \sum_i d_i$, we are considering the problem

$$\max_{y \in \{-1,1\}^V} y^T Qy, \tag{13}$$

where Q is an arbitrary symmetric matrix with zeros on the diagonal. Apart from the restriction to symmetric matrices, this is the same family of quadratic programs studied by Charikar and Wirth [CW04]. It helps intuition, however, to continue to think about $A = -Q$ as the adjacency matrix of a weighted undirected graph.

We define the *gain ratio* of a graph as the quantity

$$\gamma(G) := \max_{y \in \{-1,0,1\}^V} -\frac{y^T Ay}{y^T Dy}. \tag{14}$$

In the *gain ratio*, we consider all subsets $S \subseteq V$ of vertices and all partitions $(L, R = S - L)$ of the set S; the objective function is the ratio between twice the difference of cut edges minus uncut edges among the edges induced by S, divided by the volume of S. If one imposed the additional constraint $S = V$, then one would recover the Max Cut Gain problem.

Let λ_n be the smallest eigenvalue of the matrix $M := D^{-1/2}AD^{-1/2}$; then we see that

$$\gamma(G) \leq |\lambda_n| \tag{15}$$

because

$$|\lambda_n| = -\min_{z \in \mathbb{R}^V} \frac{z^T Mz}{z^T z} = \max_{x \in \mathbb{R}^V} -\frac{x^T Ax}{x^T Dx} \geq \max_{y \in \{-1,0,1\}^V} -\frac{y^T Ay}{y^T Dy} = \gamma(G).$$

In light of the results of Bilu and Linial, it would seem plausible to conjecture that

$$\gamma(G) \geq \Omega\left(\frac{|\lambda_n|}{\log \frac{1}{|\lambda_n|}}\right), \tag{16}$$

but we are only able to prove the considerably weaker result that $\gamma(G) \geq e^{-O(1/|\lambda_n|)}$. Subsequent to our work, Bhaskara, Charikar, and Vijayaraghavan [BCV09] proved that the conjecture of (16) is false and that there is a family of graphs in which the γ is inverse exponentially small in $1/\||\lambda_n|$.

We use the following approach. Let $x \in \mathbb{R}^V$ be a real vector and Y be a distribution over discrete vectors $\{-1,0,1\}^V$. We say that Y is a (c_1, c_2, δ)-good (randomized) rounding of x if

1. $|c_1 \cdot \mathbb{E}\, Y_iY_j - x_ix_j| \leq \delta \cdot (x_i^2 + x_j^2)$,
2. $\mathbb{E}\,|Y_i| \leq c_2 x_i^2$.

(Arguing about the "closeness" of x_ix_j to $\mathbb{E}\, Y_iY_j$ is similar to the argument used in the analysis of the rounding algorithm for the SDP relaxation of Max Cut Gain.)

We have the following simple fact.

CLAIM 6. *If x is a vector such that $-x^TAx \geq \varepsilon \cdot x^TDx$, and Y is a (c_1, c_2, δ)-good rounding of x, then the support of Y contains a vector $y \in \{-1,0,1\}^V$ such that*

$$-y^TAy \geq \frac{1}{c_1c_2}(\varepsilon - 2\delta) \cdot y^TDy.$$

Proof. We have

$$\begin{aligned}
&\mathbb{E}\sum_{i,j} -A_{ij}Y_iY_j \\
&\geq \frac{1}{c_1}\left(\sum_{ij} -A_{i,j}x_ix_j\right) + \frac{1}{c_1}2\delta\sum_i d_ix_i^2 \\
&\geq \frac{1}{c_1} \cdot (\varepsilon - 2\delta)\sum_i d_ix_i^2 \\
&\geq \frac{1}{c_1c_2} \cdot (\varepsilon - 2\delta)\,\mathbb{E}\sum_i d_i|Y_i|,
\end{aligned}$$

and so

$$\frac{\mathbb{E}\sum_{i,j} -A_{ij}Y_iY_j}{\mathbb{E}\sum_i d_i|Y_i|} \geq \frac{1}{c_1c_2}(\varepsilon - 2\delta),$$

and, in particular, there must exist a vector $y \in \{-1,0,1\}$ such that

$$\frac{\sum_{i,j} -A_{ij}y_iy_j}{\sum_i d_i|y_i|} \geq \frac{1}{c_1c_2}(\varepsilon - 2\delta). \qquad \square$$

LEMMA 7 (main). *For every $x \in \mathbb{R}^V$ and every $\ell > 1$ there is a $(c_1, c_2, 1/\ell)$-good rounding of x such that $c_1 \cdot c_2 \leq \ell^{-1} \cdot e^{\ell}$.*

Proof. Given x, we assume without loss of generality that $|x_i| \leq 1$ for every i, and we consider the following distribution Y:

- Pick a threshold $t \in [0,1]$ so that t^2 is uniformly distributed in $[0,1]$.
- For every vertex i, pairwise independently:
  - If $|x_i| > t$ or $|x_i| < t \cdot e^{-\ell}$, then set $Y_i := 0$.
  - If $t \cdot e^{-\ell} \le |x_i| \le t$, then set $Y_i := sign(x_i)$ with probability $|x_i|/t$, and $Y_i := 0$ with probability $1 - |x_i|/t$.

We begin with the calculation of the expectations $\mathbb{E}\,|Y_i|$.

Claim 8. $\mathbb{E}\,|Y_i| = 2 \cdot (e^\ell - 1) \cdot x_i^2$.

Proof of Claim 8. The threshold t is chosen according to a distribution whose density function is $2t$ for $t \in [0,1]$; conditioned on a specific choice of t, the expectation of $|Y_i|$ is 0 if $|x_i| > t$ or $|x_i| < te^{-\ell}$, and it is $|x_i|/t$ otherwise. Hence, we have

$$\mathbb{E}\,|Y_i| = \int_{|x_i|}^{|x_i|e^\ell} 2t \cdot \frac{|x_i|}{t} dt = \int_{|x_i|}^{|x_i|e^\ell} 2|x_i| dt = 2 \cdot (e^\ell - 1) \cdot x_i^2. \qquad \square$$

Claim 8 tells us that we can take $c_2 = 2 \cdot (e^\ell - 1) \le 2e^\ell$. The following two claims give us that we can take $c_1 = 1/2\ell$, so that $c_1 c_2 \le \frac{1}{\ell} \cdot e^\ell$, as required.

Claim 9. *If $|x_i| > e^\ell |x_j|$, then, for every c,*

$$|c\,\mathbb{E}\,Y_iY_j - x_ix_j| \le \frac{1}{\ell} x_i^2.$$

Proof of Claim 9. Just note that, under the assumption of the claim, $\mathbb{E}\,Y_iY_j = 0$ and $|x_ix_j| \le e^{-\ell}x_i^2 \le \ell^{-1}x_i^2$. $\square$

Claim 10. *If $|x_j| \le |x_i| \le e^\ell |x_j|$, then*

$$\left| \frac{1}{2\ell} \cdot \mathbb{E}\,Y_iY_j - x_ix_j \right| \le \frac{1}{\ell} \cdot x_i^2.$$

Proof of Claim 10. Consider the expectation of Y_iY_j, $i \ne j$, conditioned on a fixed choice of t.

$Y_iY_j = 0$ whenever $|x_i| \ge t$ or $|x_j| \le te^{-\ell}$. If t is such that $|x_i| \le t \le |x_j|e^\ell$, then the conditional expectation of Y_iY_j is x_ix_j/t^2. Overall, we have

$$\mathbb{E}\,Y_iY_j = \int_{|x_i|}^{|x_j|e^\ell} 2t \cdot \frac{x_ix_j}{t^2} dt = 2x_ix_j \cdot \int_{|x_i|}^{|x_j|e^\ell} \frac{1}{t} dt = 2x_ix_j \cdot \left(\ell - \ln\frac{|x_i|}{|x_j|}\right).$$

So we have

$$\begin{aligned}\left| \frac{1}{2\ell} \cdot \mathbb{E}\,Y_iY_j - x_ix_j \right| &= |x_ix_j| \cdot \frac{1}{\ell} \cdot \ln\frac{|x_i|}{|x_j|} \\ &= x_i^2 \cdot \frac{1}{\ell} \cdot \frac{|x_j|}{|x_i|} \cdot \ln\frac{|x_i|}{|x_j|} \\ &\le x_i^2 \cdot \frac{1}{\ell},\end{aligned}$$

where the last inequality follows from that $\rho \ln \frac{1}{\rho} \le 1$ for every $0 < \rho \le 1$. $\square$

The lemma now follows. $\square$

In order to make the proof constructive, we need to show that we can find a vector y in the sample space of Y as in the conclusion of the lemma. Suppose that the distribution of Y described above is such that $-\,\mathbb{E}\,Y^TAY \ge \mathbb{E}\,\delta Y^TDY$.

A first observation is that there must be a threshold t^* such that, conditioned on that particular choice of t, we still have $-\,\mathbb{E}[Y^TAY|t = t^*] \ge \delta\,\mathbb{E}[Y^TDY|t = t^*]$.

Once we find such a threshold, we can search in the sample space of $Y|t = t^*$, which is of polynomial size.

It remains to describe how to find a threshold t^* as above. Let us say that two thresholds t_1, t_2 are *combinatorially indistinguishable* if the sets of vertices $\{i : \delta t_1 \leq |x_i| \leq t_1\}$ and $\{i : \delta t_2 \leq |x_i| \leq t_2\}$ are equal, and call S the set of vertices.

Then we have

$$-\frac{\mathbb{E}[Y^T AT|t = t_1]}{\mathbb{E}[Y^T DY|t = t_1]} = -\frac{\sum_{i,j\in S} A_{ij} x_i x_i / t_1^2}{\sum_{i\in S} d_i |x_i| / t_1} = -\frac{1}{t_1} \cdot \frac{\sum_{i,j\in S} A_{ij} x_i x_i}{\sum_{i\in S} d_i |x_i|},$$

and, similarly,

$$-\frac{\mathbb{E}[Y^T AT|t = t_2]}{\mathbb{E}[Y^T DY|t = t_2]} = -\frac{1}{t_2} \cdot \frac{\sum_{i,j\in S} A_{ij} x_i x_i}{\sum_{i\in S} d_i |x_i|}$$

so that it is always preferable to choose the smaller threshold. This means that for every equivalence class of combinatorially indistinguishable thresholds we need only look at one of them in order to find t^*, and so we need only consider at most $2|V|$ thresholds. In particular, t^* can be found in $O(|E| + |V|)$ time. A nearly pairwise independent sample space of size $\tilde{O}(|V|)$ can be used instead of a perfectly pairwise independent one so that the whole algorithm takes time $\tilde{O}(|V| + |E|)$, at the price of an $o(1)$ additive loss in the approximation.

The following theorem summarizes our progress so far.

THEOREM 11. *There is a nearly quadratic time algorithm that, given as input a graph $G = (V, E)$ such that $\gamma(G) \geq \varepsilon$, finds a set S and a partition (L, R) of S whose gain is at least $e^{-\Omega(1/\varepsilon)}$.*

Proof. We call the algorithm *Four-Threshold Spectral Cut*, or 4TSC.

Algorithm 4TSC.

```
• Input: Graph $G = (V, E)$.
    – Let $A$ be the adjacency matrix of $G$, let $D$ be the matrix of degrees,
      and let $M := D^{-1/2} M D^{-1/2}$. Find a vector $x \in \mathbb{R}^V$ such that $\varepsilon :=$
      $-x^T Mx / x^T x \leq 2|\lambda_n|$, where $\lambda_n$ is the smallest eigenvalue of $M$. Set
      $\ell = 10/\varepsilon$.
    – For every threshold $t$ in the set $\{x(i) : i \in V\} \cup \{e^{-\ell} x(i) : i \in V\}$:
        * Let $D_t$ be a distribution over $\{-1, 0, +1\}^n$ such that when we sample
          $(Y_1, \ldots, Y_n) \sim D_t$ the $Y_i$ are $\varepsilon/10$-close to pairwise independence,
          and such that, for every $i = 1, \ldots, n$,
            1. $\Pr_{(Y_1,\ldots,Y_n)\sim D_t}[Y_i = 0] = 1$ if $|x_i| > t$ or $|x_j| < e^{-\ell} t$;
            2. $\Pr_{(Y_1,\ldots,Y_n)\sim D_t}[Y_i = sign(x_i)] = |x_i|/t$ otherwise.
          And let $\Omega_t$ be the set of vectors $(y_1, \ldots, y_n)$ having positive proba-
          bility in the distribution $D_t$
    – Output the vector $y$ in the union of $\Omega_t$ that maximizes $\frac{\sum_{i,j} A_{i,j}|y_i + y_j|}{\sum_i d_i |y_i|}$.
```

Using the construction of almost pairwise independent random variables of Alon et al. [AGHP92], each sample space Ω_t has size $\tilde{O}(\log n)$ and can be computed in $\tilde{O}(n)$ time. For each vector y, the ratio can be computed in linear time. □

By iterating the algorithm we derive our main result of this section.

THEOREM 12. *There is a nearly cubic time algorithm that, given as input a graph $G = (V, E)$ such that $max - cut - gain(G) \geq \varepsilon$, finds a cut (L, R) of V of gain $\geq e^{-\Omega(1/\varepsilon)}$.*

8. Conclusions.

The motivating question for this work was to find a combinatorial interpretation of the quantity $d - |\lambda_n|$ in a d-regular graph, akin to the interpretation of $d - \lambda_2$ provided by the theory of edge expansion.

In establishing such an interpretation (in terms of the quantity that we call the "bipartiteness ratio" in section 5) we proved that a natural and easy-to-implement spectral algorithm performs nontrivially well with respect to the Max Cut problem.

The algorithm is very fast in practice [OT08]; using a termination rule that is slightly more relaxed than that used in this paper (stopping when $U + X > M/2$, instead of $U + X/2 > M/2$), the algorithm makes at most one recursive call in all the experiments that we performed. It would be interesting to give a proof that this is always the case.

A number of interesting open questions remain:

1. What is the worst-case approximation ratio of our algorithm? Soto [Sot09] improves our analysis and shows than an approximation ratio of at least .614 is always achieved, and perhaps Soto's analysis can be improved further.
2. Is there a "purely combinatorial" algorithm (namely, one not involving numerical matrix computations) for Max Cut achieving an approximation factor better than 1/2?
3. The counterexample of Bhaskara, Charikar, and Vijayaraghavan [BCV09] shows that our analysis for Max Cut Gain cannot be improved significantly. Is there an algorithm not based on the rounding of an SDP that, given a graph in which the Max Cut optimum is $1/2 + \varepsilon$, finds a cut of cost at least $1/2 + \varepsilon^{O(1)}$?

Acknowledgments. I would like to thank an anonymous referee for asking the question regarding the connection between spectral techniques and Max Cut, and Sebastian Cioaba, Satyen Kale, James Lee, Vladimir Nikiforov, and Salil Vadhan for providing helpful comments and references to the related literature.

I am grateful to Moses Charikar for communicating the proof of Theorem 4, which substantially improved my previous analysis, and for allowing me to present his improved analysis in this paper.

REFERENCES

[ACMM05] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev, $O(\sqrt{\log n})$ *approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems*, in Proceedings of the 37th ACM Symposium on Theory of Computing, ACM, New York, 2005, pp. 573–581.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta, *Simple constructions of almost k-wise independent random variables*, Random Structures Algorithms, 3 (1992), pp. 289–304.

[AK07] S. Arora and S. Kale, *A combinatorial, primal-dual approach to semidefinite programs*, in Proceedings of the 39th ACM Symposium on Theory of Computing, ACM, New York, 2007, pp. 227–236.

[Alo86] N. Alon, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[AN06] N. Alon and A. Naor, *Approximating the cut-norm via Grothendieck's inequality*, SIAM J. Comput., 35 (2006), pp. 787–803.

[ARV09] S. Arora, S. Rao, and U. V. Vazirani, *Expander flows, geometric embeddings and graph partitioning*, J. ACM, 56 (2009), 5:1–5:37.

[AS00] N. Alon and B. Sudakov, *Bipartite subgraphs and the smallest eigenvalue*, Combin. Probab. Comput., 9 (2000), pp. 1–12.

[BCV09] A. Bhaskara, M. Charikar, and A. Vijayaraghavan, *On the Eigenvalue Relaxation for Max Cut Gain*, preprint, Princeton University, Princeton, NJ, 2009.

[BL06] Y. Bilu and N. Linial, *Lifts, discrepancy and nearly optimal spectral gap*, Combinatorica, 26 (2006), pp. 495–519.

[Bra98] S. Brandt, *The local density of triangle-free graphs*, Discrete Math., 183 (1998), pp. 17–25.

[CW04] M. Charikar and A. Wirth, *Maximizing quadratic programs: Extending Grothendieck's inequality*, in Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 2004, pp. 54–60.

[dlVKM07] W. F. de la Vega and C. Kenyon-Mathieu, *Linear programming relaxations of maxcut*, in Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2007, pp. 53–61.

[DP93a] C. Delorme and S. Poljak, *Combinatorial properties and the complexity of a max-cut approximation*, European J. Combin., 14 (1993), pp. 313–333.

[DP93b] C. Delorme and S. Poljak, *Laplacian eigenvalues and the maximum cut problem*, Math. Program., 62 (1993), pp. 557–574.

[GW95] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[Hae79] W. Haemers, *Eigenvalue Techniques in Design and Graph Theory*, Ph.D. thesis, Eindhoven University of Technology, 1979.

[Kho02] S. Khot, *On the power of unique* 2*-prover* 1*-round games*, in Proceedings of the 34th ACM Symposium on Theory of Computing, ACM, New York, 2002, pp. 767–775.

[KKMO04] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, *Optimal inapproximability results for MAX-CUT and other two-variable CSPs?*, in Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 2004, pp. 146–154.

[KO06] S. Khot and R. O'Donnell, *SDP gaps and UGC-hardness for MAXCUTGAIN*, in Proceedings of the 47th IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 2006, pp. 217–226.

[KRV06] R. Khandekar, S. Rao, and U. V. Vazirani, *Graph partitioning using single commodity flows*, in Proceedings of the 38th ACM Symposium on Theory of Computing, ACM, New York, 2006, pp. 385–390.

[KW92] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.

[Lov03] L. Lovász, *Semidefinite programs and combinatorial optimization*, in Recent Advances in Algorithms and Combinatorics, B. A. Reed and C. L. Linhares-Sales, eds., Springer, New York, 2003, pp. 137–194.

[MOO10] E. Mossel, R. O'Donnell, and K. Oleszkiewicz, *Noise stability of functions with low influences: Invariance and optimality*, Ann. of Math., 171 (2010), pp. 295–341.

[Nik07] V. Nikiforov, *The spectral radius of subgraphs of regular graphs*, Electron. J. Combin., 14 (2007), p. N20.

[OSVV08] L. Orecchia, L. J. Schulman, U. V. Vazirani, and N. K. Vishnoi, *On partitioning graphs via single commodity flows*, in Proceedings of the 40th ACM Symposium on Theory of Computing, ACM, New York, 2008, pp. 461–470.

[OT08] G. Ottaviano and L. Trevisan, *An Experimental Analysis of a Spectral Approximation Algorithm for MAX CUT*, preprint, University of California at Berkeley, Berkeley, CA, 2008.

[OW08] R. O'Donnell and Y. Wu, *An optimal SDP algorithm for Max-Cut, and equally optimal long code tests*, in Proceedings of the 40th ACM Symposium on Theory of Computing, ACM, New York, 2008, pp. 335–344.

[Sot09] J. Soto, *Improved Analysis of a Max Cut Algorithm Based on Spectral Partitioning*, preprint, http://arxiv.org/abs/0910.0504, 2009.

[ST04] D. Spielman and S.-H. Teng, *Nearly linear time algorithms for graph partitioning, graph sparsification, and solving linear systems*, in Proceedings of the 36th ACM Symposium on Theory of Computing, ACM, New York, 2004, pp. 81–90.

[STT07] G. Schoenebeck, L. Trevisan, and M. Tulsiani, *Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut*, in Proceedings of the 39th ACM Symposium on Theory of Computing, ACM, New York, 2007, pp. 302–310.

[Tre08] L. Trevisan, *Max Cut and the Smallest Eigenvalue*, preprint, http://arxiv.org/abs/0806.1978, 2008.