ON THE QUALITY OF SPECTRAL SEPARATORS*

STEPHEN GUATTERY† AND GARY L. MILLER‡

Abstract. Computing graph separators is an important step in many graph algorithms. A popular technique for finding separators involves spectral methods. However, there has not been much prior analysis of the quality of the separators produced by this technique; instead it is usually claimed that spectral methods "work well in practice." We present an initial attempt at such analysis. In particular, we consider two popular spectral separator algorithms, and provide counterexamples that show these algorithms perform poorly on certain graphs. We also consider a generalized definition of spectral methods that allows the use of some specified number of the eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix of a graph, and show that if such algorithms use a constant number of eigenvectors, then there are graphs for which they do no better than using only the second smallest eigenvector. Further, using the second smallest eigenvector of these graphs produces partitions that are poor with respect to bounds on the gap between the isoperimetric number and the cut quotient of the spectral separator. Even if a generalized spectral algorithm uses n^ϵ for $0 < \epsilon < \frac{1}{4}$ eigenvectors, there exist graphs for which the algorithm fails to find a separator with a cut quotient within $n^{\frac{1}{4}-\epsilon} - 1$ of the isoperimetric number. We also introduce some facts about the structure of eigenvectors of certain types of Laplacian and symmetric matrices; these facts provide the basis for the analysis of the counterexamples. Finally, we discuss some developments in spectral partitioning that have occurred since these results first appeared.

Key words. graph partitioning, spectral partitioning, graph eigenvalues and eigenvectors

AMS subject classifications. 05C50, 05C85, 15A18, 68Q25, 68R10

1. Introduction. Spectral methods (i.e., methods that use the eigenvalues and eigenvectors of a matrix representation of a graph) are widely used to compute graph separators. Typically, the Laplacian matrix is used; the Laplacian B of a graph G on n vertices is the $n \times n$ matrix with the degrees of the vertices of G on the diagonal, and entry $b_{ij} = -1$ if G has the edge (v_i, v_j) and 0 otherwise. The eigenvector $\mathbf{u}_2$ corresponding to λ_2 (the second-smallest eigenvalue of B) is computed, and the vertices of the graph are partitioned according to the values of their corresponding entries in $\mathbf{u}_2$ [23, 18]. The goal is to compute a small separator; that is, as few edges or vertices as possible should be deleted from the graph to achieve the partition. Additionally, the sizes of the resulting components should be roughly comparable.

Although spectral methods are popular, there has been little previous analysis of the quality of the separators they produce. Instead, it is often claimed that such methods "work well in practice," and tables of results for specific examples are often included in papers (see e.g. [23]). Thus there is little guidance for someone wishing to compute separators as to whether or not this technique is appropriate. Ideally, practitioners should have a characterization of classes of graphs for which spectral separator techniques work well; this characterization might be in terms of how far the computed separators can be from optimal. This paper represents a first step in this direction. We consider two spectral separation algorithms that partition the vertices on the basis of the values of their corresponding entries in $\mathbf{u}_2$, and provide

*An extended abstract of this paper appeared in the Sixth Annual ACM/SIAM Symposium on Discrete Algorithms (SODA '95). This paper is a revised version of CMU Computer Science Technical Report CMU-CS-94-228.

†ICASE, Mail Stop 403, NASA Langley Research Center, Hampton, VA 23681 (smg@icase.edu). This work supported in part by NAS contract NAS1-19480.

‡School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (glmiller@cs.cmu.edu). This work supported in part by NSF Grant CCR-9505472.

counterexamples from classes of practical interest for which each of the algorithms produces poor separators. We further consider a generalized definition of spectral methods that allows the use of more than one of the eigenvectors corresponding to the smallest non-zero eigenvalues, and show that there are graphs for which any such algorithm does poorly.

The first algorithm bisects a graph by partitioning the vertices into two equal-sized sets based on each vertex's entry in the eigenvector $\mathbf{u}_2$. The class of bounded-degree planar counterexamples for this method consists of graphs that look like ladders with the top 1/2 of their rungs kicked out; a straightforward spectral bisection algorithm cuts the remaining rungs, whereas the optimal bisection is made by cutting across the ladder above the remaining rungs. The counterexample graphs have $\Theta(1)$ bisectors; the spectral bisection algorithm produces a $\Theta(n)$ bisection, which is as far from the optimum as possible (to within a constant).

The spectral bisection algorithm can be modified to generate a better separator for the bisection counterexample. Some modifications are presented in [18]; they still use a partition based on $\mathbf{u}_2$. We consider a simple spectral separator algorithm, the "best threshold cut" algorithm, based on the most general of these suggested modifications. (In such an algorithm, "best" is measured in terms of the **cut quotient**, the ratio between the number of edges cut and the size of the smaller set in the vertex partition; the smallest cut quotient over all separators is called the **isoperimetric number**.) We present a class of graphs that defeats this algorithm in that the ratio of the spectral cut's cut quotient to the isoperimetric number is as bad as possible (to within a constant) with respect to bounds on these quantities.

We also consider a more general definition of purely spectral separator algorithms that subsumes the two preceding algorithms. This definition allows the use of some specified number of eigenvectors corresponding to the smallest eigenvalues of the Laplacian. For any such algorithm that uses a fixed number of eigenvectors we show there are graphs for which it does no better than using the "best threshold cut" algorithm. Further, the separator produced when the "best threshold cut" algorithm is applied to these graphs is as bad as possible (to within a constant) with respect to bounds on the size of the separators produced. We also show that if a purely spectral algorithm uses up to n^ϵ eigenvectors for $0 < \epsilon < \frac{1}{4}$, there exist graphs for which the algorithm fails to find a separator with a cut quotient within a factor of $n^{\frac{1}{4}-\epsilon} - 1$ times the isoperimetric number.

Finally, we provide a summary of some important subsequent results by Spielman and Teng [27], and relate our results to them.

This paper makes an additional contribution: While the counterexamples have simple structures and intuitively might be expected to cause problems for spectral separator algorithms, the challenge is to provide good bounds on λ_2 for these graphs. For this purpose we have developed theorems about the spectra of graphs with particular symmetries (i.e., automorphisms of order 2) that exist in the counterexamples.

Specifics are given in the text that follows: Section 2 gives a brief history of spectral methods and the details of the algorithms discussed in this paper. Graph and matrix terminology and notation are presented in Section 3, which also presents some useful facts about Laplacians. Results about the eigenvalues and eigenvectors of Laplacians of graphs with automorphisms of order 2 are in Section 4. Section 5 gives the counterexample for the spectral bisection algorithm; Section 6 gives the counterexample for the "best threshold cut" algorithm. Section 7 discusses the generalized definition of spectral separator algorithms, and shows that there are graphs for which any such algorithm performs poorly. Section 8 discusses Spielman and Teng's results.

2. Spectral Methods for Computing Separators. The roots of spectral partitioning go back to Hoffmann and Donath [9], who proved a lower bound on the size of the minimum bisection of a graph, and Fiedler [11][12], who explored the properties of λ_2 and its associated eigenvector for the Laplacian. There has been much subsequent work, including Barnes's partitioning algorithm [5], Boppana's work that included a stronger lower bound on the minimum bisection size [6], work by Rendl, Wolkowicz, and others using optimization approaches [24, 10], and the particular bisection and graph partitioning methods considered in this paper [18] [23] [25]. Since our work first appeared [17], Spielman and Teng [27] have extended the latter methods to include recursion. (It is worth noting that spectral methods have not been limited to graph partitioning; work has been done using the spectrum of the adjacency matrix in graph coloring [4] and using the Laplacian spectrum to prove theorems about expander graph and superconcentrator properties [3] [1] [2]. The work on expanders has explored the relationship of λ_2 to the isoperimetric number; Mohar has given an upper bound on the isoperimetric number using a strong discrete version of the Cheeger inequality [22]. Reference [8] is a book-length treatment of graph spectra, and it predates many of the results cited above.)

A basic way of computing a graph bisection using spectral information is presented in [23]. We refer to this algorithm as **spectral bisection**. Spectral bisection works as follows:

- Represent G by its Laplacian B, and compute $\mathbf{u}_2$, the eigenvector corresponding to λ_2 of B.
- Assign each vertex the value of its corresponding entry in $\mathbf{u}_2$. This is the **characteristic valuation** of G.
- Compute the median of the elements of $\mathbf{u}_2$. Partition the vertices of G as follows: the vertices whose values are less than or equal to the median form one part; the rest of the vertices form the other part. The set of edges between the two parts forms an edge separator.
- If a vertex separator is desired, it is computed from the edge separator using standard techniques described in the next section.

Since the graph bisection problem is NP-complete [13], spectral bisection may not give an optimum result. That is, spectral bisection is a heuristic method. A number of modifications have been proposed that may improve its performance. These modified heuristics may give splits other than bisections. In such cases, one can use the cut quotient to judge the quality of the split. Computing a separator with a cut quotient equal to the isoperimetric number is NP-hard [14]. The following modifications, all of which use the characteristic valuation, are presented in [18]:

- Partition the vertices based on the signs of their values;
- Look for a large gap in the sorted list of eigenvector components, and partition the vertices according to whether their values are above or below the gap; and
- Sort the vertices according to value. For each index $1 \le i \le n-1$, consider the ratio for the separator produced by splitting the vertices into those with sorted index $\le i$ and those with sorted index $> i$. Choose the split that provides the best cut quotient.

Note that the last idea subsumes the first two. We consider a variant of this algorithm below. Since this algorithm does not specify what to do when multiple vertices have the same value, we restrict it to consider only splits between vertices with different values (such cuts are called **threshold cuts**). This restricted version is the **"best threshold cut" algorithm**; the slight change from the definition above does not alter its performance with respect to the counterexamples below (other than slightly simplifying the analysis).

Also note that the idea of cutting at an arbitrary point along the sorted order can be extended to choosing two split points, where the corresponding partitions are the vertices with values between the split points, and those with values above the upper or below the lower split point. Again, the pair yielding the best ratio is chosen.

The algorithms mentioned so far have only used the eigenvector $\mathbf{u}_2$. Another possibility is to look at partitions generated by the set of eigenvectors for some number of smallest eigenvalues: for each vertex, a value is assigned by computing a function of that vertex's eigenvector components. Partitions are then generated in the same way as they are for $\mathbf{u}_2$ in the various algorithms given above.

Given the variety of heuristics cited above, it would be nice to know which ones work well for which classes of graphs. It would be particularly useful if it were possible to state reasonable bounds on the performance of these heuristics for classes of graphs commonly used in practice (e.g., planar graphs, planar graphs of bounded degree, three-dimensional finite element meshes, etc.). Unfortunately, this is not the case. We start by proving that spectral bisection may produce a bad separator for a bounded-degree planar graph in Section 5; first, however, we need to introduce some terminology and background results.

3. Terminology, Notation, and Background Results. We assume that the reader is familiar with the basic definitions of graph theory (in particular, for undirected graphs), and with the basic definitions and results of matrix theory. A graph consists of a set of vertices V and a set of edges E; we denote the vertices (respectively edges) of a particular graph G as $V(G)$ (respectively $E(G)$) if there is any ambiguity about which graph is referred to. The notation $|G|$ is sometimes be used as a shorthand for $|V(G)|$. When it is clear which graph is referred to, we use n to denote $|V|$.

Capital letters represent matrices and bold lower-case letters represent vectors. For a matrix A, a_{ij} or $[A]_{ij}$ represents the element in row i and column j; for the vector $\mathbf{x}$, x_i or $[\mathbf{x}]_i$ represents the i^{th} entry in the vector. The notation $\mathbf{x} = 0$ indicates that all entries of the vector $\mathbf{x}$ are zero; $\vec{1}$ indicates the vector that has 1 for every entry. For ease of reference, the eigenvalues of an $n \times n$ matrix are indexed in non-decreasing order. λ_1 represents the smallest eigenvalue, and λ_n the largest. For $1 < i < n$, $\lambda_{i-1} \leq \lambda_i \leq \lambda_{i+1}$. The notation $\lambda_i(A)$ (respectively $\lambda_i(G)$) indicates the i^{th} eigenvalue of matrix A (respectively of the Laplacian of graph G) if there is any ambiguity about which matrix (respectively graph) the eigenvalue belongs to. We use $\mathbf{u}_i$ to represent the eigenvector corresponding to λ_i.

A **path graph** is a tree with exactly two vertices of degree one.

The **crossproduct** of two graphs G and H (denoted $G \times H$) is a graph on the vertex set $\{(u, v) \mid u \in V(G), v \in V(H)\}$, with $((u, v), (u', v'))$ in the edge set if and only if either $u = u'$ and $(v, v') \in E(H)$ or $v = v'$ and $(u, u') \in E(G)$. It is easy to see that $G \times H$ and $H \times G$ are isomorphic. One way to think of a graph crossproduct is as follows: Replace every vertex in G with a copy of H. Each edge e in G is then replaced by $|H|$ edges, one between each pair of corresponding vertices in the copies of H that have replaced the endpoints of e. An example is shown in Figure 3.1.

For a connected graph G, an **edge separator** is a set S of edges that, if removed, breaks the graph into two (not necessarily connected) components G_1 and G_2 that

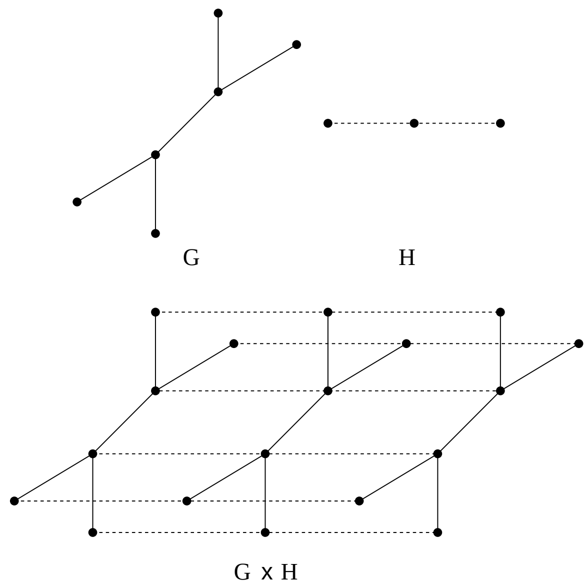

FIG. 3.1. *A Graph Crossproduct Example*

have no edges between them. (An edge separator is by definition minimal with respect to G_1 and G_2.) A **vertex separator** is a set S of vertices such that if these vertices and all incident edges are removed, the graph is broken into two components G_1 and G_2 that have no edges between them (again, a separator is a minimal such set). The goal in finding separators is to find a small separator that breaks the graph into two fairly large pieces; often this notion is expressed as a balance restriction that requires the number of vertices in each of G_1 and G_2 to be at least some specified fraction of the number of vertices in G. For edge separators, this goal is stated more generally in terms of minimizing some measure relating the size of the separator to the size of the resulting components. One such measure that we use is the **isoperimetric number** $i(G)$, defined as:

$$\min_S \left(\frac{|S|}{\min\left(|G_1|, |G_2|\right)} \right).$$

We refer to the quantity $|S|/\min\left(|G_1|, |G_2|\right)$ as the **cut quotient** for the edge separator S. As noted Section 2, finding a cut with a cut quotient equal to the isoperimetric number is NP-hard. It is well known that an edge separator S can be converted into a vertex separator S' by considering the bipartite graph induced by S and setting S' to be a minimum vertex cover for that graph.

Given a vertex numbering, graphs can be represented by matrices. For example, the **adjacency matrix** A of a graph G is defined as $a_{ij} = 1$ if and only if $(v_i, v_j) \in E(G)$; $a_{ij} = 0$ otherwise. A common matrix representation of graphs is the **Laplacian**. Let D be the matrix with $d_{ii} = \text{degree}(v_i)$ for $v_i \in V(G)$, and all off-diagonal entries equal to zero. Let A be the adjacency matrix for G. Then the Laplacian of G is the matrix $B = D - A$.

The following are useful facts about the Laplacian matrix:

- The Laplacian is symmetric positive semidefinite (see e.g. [21]).

- A graph G is connected if and only if 0 is a simple eigenvalue of its Laplacian (see e.g. [21]). The eigenvector for 0 is $\vec{1}$.
- The following characterization of λ_2 holds (see e.g. [11]):

$$\lambda_2 = \min_{\mathbf{x} \perp \vec{1}} \frac{\mathbf{x}^T B \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.$$

- If G is a crossproduct of two graphs G and H, then the eigenvalues of the Laplacian of G are all pairwise sums of the eigenvalues of G and H (see e.g. [21]).
- For any vector $\mathbf{x}$ and Laplacian B of graph G, the following holds (see e.g. [18]):

$$\mathbf{x}^T B \mathbf{x} = \sum_{(v_i, v_j) \in E(G)} (x_i - x_j)^2 \tag{3.1}$$

- For a graph G that is not one of K_1, K_2, or K_3 (the complete graphs on 1, 2, and 3 vertices respectively), let λ_2 be the smallest nonzero eigenvalue of its Laplacian. G's isoperimetric number can be bounded as follows [22]:

$$\frac{\lambda_2}{2} \leq i(G) \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}, \tag{3.2}$$

  where Δ is the maximum degree of any vertex in G.

The proof of the upper bound in (3.2) has interesting implications about the threshold cuts based on the second eigenvector. For any connected graph G, consider the characteristic valuation. The vertices of G receive $k \leq n$ distinct values; let $t_1 > t_2 > \ldots > t_k$ be these values. For each threshold t_i, $i < k$, divide the vertices into those with values greater than t_i, and those with values less than or equal to t_i. Compute the cut quotient q_i for each such cut, and let q_{min} be the minimum over all q_i's. The following theorem can be derived from the proof of Theorem 4.2 in [22] (a similar argument leading to similar result for the Laplace operator associated with the transition matrix of a reversible Markov chain can be found in [26]):

THEOREM 3.1. *Let G be a connected graph with maximal vertex degree Δ and second smallest eigenvalue λ_2. If G is not any of K_1, K_2, or K_3, then*

$$\frac{\lambda_2}{2} \leq q_{min} \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}.$$

A **weighted** graph is a graph for which a real value w_i is associated with each vertex v_i, and a real, nonzero weight w_{ij} is associated with each edge (v_i, v_j) (a zero edge weight indicates the lack of an edge). Fiedler extended the notion of the Laplacian to graphs with positive edge weights [12]; he referred to this representation as the **generalized Laplacian**. Our results require a representation for graphs with vertex weights and negative edge weights. Hence we define the **standard matrix representation** B of a weighted graph G as follows: B has $b_{ii} = w_i$; for $i \neq j$ and $(v_i, v_j) \in E(G)$, $b_{ij} = -w_{ij}$, and $b_{ij} = 0$ otherwise. Note that the standard matrix representation of any weighted graph is a real symmetric matrix, and that any such matrix can be represented as a specific weighted graph. Note also that the Laplacian matrix of a graph is also the standard matrix representation of the graph with vertex weights equal to the vertex degrees, and all edge weights set to 1.

4. Automorphisms of Order 2 and Eigenvector Structure. The theorems and lemmas presented in this section are useful in proving results about the eigenvectors of the families of graphs presented in later sections. The details of the proofs are not necessary to understanding the rest of the paper; a reader interested only in understanding the counterexamples and their implications can look at the theorem statements and skip the proofs.

The first set of results concerns eigenvalues of Laplacians of graphs with automorphisms of order 2. A **graph automorphism** is a permutation ϕ on the vertices of the graph G such that $(v_i, v_j) \in E(G)$ if and only if $(v_{\phi(i)}, v_{\phi(j)}) \in E(G)$. The **order** of a graph automorphism is the order of the permutation ϕ, the minimum number of times ϕ must be applied to yield the identity mapping.

For weighted graphs, there are two additional conditions: the weights of vertices v_i and $v_{\phi(i)}$ must be equal for all i, and the weights of edges (v_i, v_j) and $(v_{\phi(i)}, v_{\phi(j)})$ must be equal.

The next two theorems concern the structure of eigenvectors with respect to automorphisms of order 2. They hold both for Laplacians of graphs under the standard definition of automorphism, and for standard matrix representations of weighted graphs under the definition of automorphisms for weighted graphs.

Let G be a graph with an automorphism ϕ of order 2 and Laplacian B. A vector $\mathbf{x}$ that has $x_i = x_{\phi(i)}$ for all i in the range $1 \leq i \leq n$ is an **even** vector with respect to the automorphism ϕ; an **odd** vector $\mathbf{y}$ has $y_i = -y_{\phi(i)}$ for all i. It is easy to show that for any even vector $\mathbf{x}$ and odd vector $\mathbf{y}$ (both with respect to ϕ), $\mathbf{x}$ and $\mathbf{y}$ are orthogonal.

THEOREM 4.1. *Let B be the Laplacian of a graph G that has an automorphism ϕ of order 2. Then there exists a complete set $\mathcal{U}$ of orthogonal eigenvectors of B such that any eigenvector $\mathbf{u} \in \mathcal{U}$ is either even or odd with respect to ϕ. This also holds if G is a weighted graph, B the standard matrix representation of G, and ϕ a weighted graph automorphism of order 2.*

Proof. Let P be the permutation matrix that corresponds to the automorphism ϕ. Then $P^T BP = B$. Let $\mathbf{u}$ be an eigenvector of B with eigenvalue λ. We have

$$\left(P^T BP\right)\mathbf{u} = B\mathbf{u} = \lambda\mathbf{u}. \tag{4.1}$$

Since the automorphism is of order 2, $PP = I$ and $P^T = P^{-1} = P$. Therefore, multiplying the left and right sides of (4.1) by P gives

$$B\left(P\mathbf{u}\right) = P\left(\lambda\mathbf{u}\right) = \lambda\left(P\mathbf{u}\right).$$

Thus, $P\mathbf{u}$ is also an eigenvector with eigenvalue λ.

Note that for an even vector $\mathbf{x}$, $P\mathbf{x} = \mathbf{x}$; for an odd vector $\mathbf{y}$, $P\mathbf{y} = -\mathbf{y}$.

P allows us to decompose any vector $\mathbf{x}$ uniquely into an odd component $\mathbf{x}_{odd}$ and an even component $\mathbf{x}_{even}$ as follows:

$$\mathbf{x}_{odd} = \frac{\mathbf{x} - P\mathbf{x}}{2}, \text{ and } \mathbf{x}_{even} = \frac{\mathbf{x} + P\mathbf{x}}{2}.$$

For any non-zero $\mathbf{x}$, at least one of the even or odd parts must also be non-zero.

Let $\mathcal{U}'$ be any complete set of eigenvectors of B. For an eigenvector $\mathbf{u} \in \mathcal{U}'$, it is easy to see that a non-zero even or odd component is an eigenvector for the same eigenvalue. Since $\mathbf{u}_{odd} + \mathbf{u}_{even} = \mathbf{u}$, the set of odd and even eigenvectors resulting from decomposing all the eigenvectors in $\mathcal{U}'$ spans the same space as $\mathcal{U}'$. The subspaces spanned by all odd and by all even components respectively are orthogonal. Since B is real and symmetric, we can subdivide these subspaces into smaller orthogonal subspaces spanned by the odd (respectively even) eigenvectors for particular eigenvalues. We can form an orthogonal basis for each of these smaller subspaces; the union of all these bases is the desired set $\mathcal{U}$ of orthogonal odd and even eigenvectors. This implies the claimed result.

The proof clearly holds whether B is a Laplacian or a standard matrix representation.

□

COROLLARY 4.2. *Let B be the standard matrix representation of a weighted graph G that has one or more automorphisms of order 2. Then the eigenvector for any simple eigenvalue is either even or odd with respect to every such automorphism.*

Proof. Let $\mathbf{u}$ be the eigenvector for some simple eigenvalue λ. Consider the decomposition of $\mathbf{u}$ into odd and even parts with respect to some automorphism ϕ with order 2. If both parts were non-zero, they would be orthogonal and eigenvectors for λ. Therefore either the odd part or the even part must be zero.

□

Since Laplacians can be considered to be standard matrix representations given the right weight assignments, the preceding result also holds for Laplacians.

Let B be a standard matrix representation of a weighted graph with an automorphism ϕ of order 2. It is possible to decompose B into two smaller matrices B_{odd} and B_{even} such that the eigenvalues of B_{odd} and B_{even} are the odd and even eigenvalues of B respectively, and further, that a full set of odd and even eigenvectors of B can be constructed in a simple way from the eigenvectors of B_{odd} and B_{even} respectively. We demonstrate this through a similarity transform based on ϕ. First, however, we need to introduce some notation.

The vertices of G can be divided into two disjoint sets on the basis of how ϕ operates on them. Let V_f be the set of vertices v_i such that $\phi(i) = i$ (i.e., the vertices fixed by ϕ), and let V_m be the set of vertices v_j such that $\phi(j) \neq j$ (i.e., the vertices moved by ϕ). V_m consists of vertices in orbits of length 2. We call a subset of V_m that consists of exactly one vertex from each such orbit a **representative set** and denote it V_r. In the rest of this presentation we assume that a particular V_r has been arbitrarily specified. We use n_f, n_m, and n_r respectively to denote the number of vertices in each of these sets.

Without loss of generality, number the vertices in the following way: the vertices in V_f are numbered 1 through n_f; the vertices in V_r are numbered from $n_f + 1$ to $n_f + n_r$. Renumber the vertices in $V_m \setminus V_r$ such that if $v_i \in V_r$, then $\phi(i) = i + n_r$; that is, the vertices in $V_m \setminus V_r$ are numbered $n_f + n_r + 1$ to n in the same order as the vertices in V_r with which they share orbits. Using this ordering and the definition of the automorphism, B can be written in the following block form:

$$B = \begin{bmatrix} F & E_{fr} & E_{fr} \\ E_{fr}^T & R & E_{r\phi(r)} \\ E_{fr}^T & E_{r\phi(r)} & R \end{bmatrix},$$

where

- F is an $n_f \times n_f$ submatrix containing the diagonal entries for the vertices in V_f and the entries for edges between pairs of vertices in V_f;
- R is an $n_r \times n_r$ submatrix containing the diagonal entries for the vertices in V_r and the entries for edges between pairs of vertices in V_r;

- E_{fr} is the entries of B for edges between vertices in V_f and V_r; and
- $E_{r\phi(r)}$ is the entries of B for edges between vertices in V_r and $V_f \setminus V_r$ (note that the conditions specified above imply $E_{r\phi(r)} = E^T_{r\phi(r)}$.

We now define the orthogonal matrix T used to transform B:

$$T = \begin{bmatrix} I_{n_f} & 0 & 0 \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{1}{\sqrt{(2)}} I_{n_r} \\ 0 & \frac{1}{\sqrt{(2)}} I_{n_r} & \frac{-1}{\sqrt{(2)}} I_{n_r} \end{bmatrix},$$

where the I's are identity matrices with the dimension specified in the subscript. B is transformed as follows:

$$B' = T^T B T = \begin{bmatrix} F & \sqrt{2}E_{fr} & 0 \\ \sqrt{2}E^T_{fr} & R + E_{r\phi(r)} & 0 \\ 0 & 0 & R - E_{r\phi(r)} \end{bmatrix}.$$

Note that the resulting matrix is reducible. That is, when viewed as a weighted graph, that graph has two components. We show that the blocks of this matrix correspond to B_{even} and B_{odd} as follows:

$$B_{even} = \begin{bmatrix} F & \sqrt{2}E_{fr} \\ \sqrt{2}E^T_{fr} & R + E_{r\phi(r)} \end{bmatrix} \quad \text{and} \quad B_{odd} = R - E_{r\phi(r)}.$$

Let B, T, B', B_{odd}, and B_{even} be as defined above.

THEOREM 4.3. *The eigenvalues of B_{odd} are odd eigenvalues of B, and a complete set of odd eigenvectors of B can be constructed from the eigenvectors of B_{odd} in a straightforward way. Likewise, the eigenvalues of B_{even} are even eigenvalues of B, and a complete set of even eigenvectors of B can be constructed from the eigenvectors of B_{even} in a straightforward way.*

Proof. Because B' is reducible, every eigenvalue of B_{odd} is an eigenvalue of B'; likewise every eigenvalue of B_{even} is an eigenvalue of B'. By similarity, they are also eigenvalues of B.

Now consider an eigenvector $\mathbf{u}$ of B_{even}. Define $\mathbf{v}$ as follows: for $1 \le i \le n_f + n_r$ let $v_i = u_i$; let $v_i = 0$ otherwise. $\mathbf{v}$ is obviously an eigenvector of B'. Multiplication by the matrix T transforms $\mathbf{v}$ into an eigenvector $\mathbf{w}$ of B:

$$\mathbf{w} = T\mathbf{v} = \begin{bmatrix} \mathbf{v}_f \\ \frac{1}{\sqrt{(2)}}\mathbf{v}_r \\ \frac{1}{\sqrt{(2)}}\mathbf{v}_r \end{bmatrix}$$

By the vertex numbering, it is easy to see this is an even vector. Since $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ all have the same eigenvalue λ, the claim about eigenvalues of B_{even} corresponding to even eigenvalues of B holds. It is easy to show that if two eigenvectors $\mathbf{u}_1$ and $\mathbf{u}_2$ of B_{even} are orthogonal, then the corresponding eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are also orthogonal. Since B_{even} has $n_f + n_r$ orthogonal eigenvectors, we have $n_f + n_r$ orthogonal even eigenvectors of B.

Now consider an eigenvector $\mathbf{u}$ of B_{odd}. As before, one can construct an eigenvector $\mathbf{v}$ of B': for $n_f + n_r + 1 \le i \le n$ let $v_i = u_i$; let $v_i = 0$ otherwise. Multiplication by the matrix T again transforms $\mathbf{v}$ into an eigenvector $\mathbf{w}$ of B:

$$\mathbf{w} = T\mathbf{v} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt(2)}\mathbf{v}_{\phi(r)} \\ \frac{-1}{\sqrt(2)}\mathbf{v}_{\phi(r)} \end{bmatrix}$$

This is clearly an odd vector. Since $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ all have the same eigenvalue λ, the claim about eigenvalues of B_{odd} corresponding to odd eigenvalues of B holds. It is easy to show that if two eigenvectors $\mathbf{u}_1$ and $\mathbf{u}_2$ of B_{odd} are orthogonal, then the corresponding eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ are also orthogonal. Since B_{odd} has n_r orthogonal eigenvectors, we have n_r orthogonal odd eigenvectors of B.

Note that if all eigenvectors of B_{even} and B_{odd} are transformed in this way the result is n orthogonal eigenvectors of B (i.e., a full set).
□

It is possible to express this decomposition in terms of graphs: The graph G is decomposed into the components G_{odd} and G_{even}. Rules for the graphical decomposition can be derived from the structure of B_{odd} and B_{even}, and are presented in the technical report version of this paper [16].

The following technical lemmas about the eigenvalues and eigenvectors of weighted path graphs are useful in subsequent results.

LEMMA 4.4. *Let B be the standard matrix representation of a weighted path graph G on n vertices. For any vector $\mathbf{x}$ such that $B\mathbf{x} = \lambda\mathbf{x}$ for some real λ, $x_n = 0$ implies $\mathbf{x} = 0$. Likewise, $x_1 = 0$ implies $\mathbf{x} = 0$. If there are two consecutive elements x_i and x_{i+1} that are both zero, then $\mathbf{x} = 0$.*

Proof. The first result is proved by induction. The base case is for a 2×2 matrix with diagonal entries b_{11} and b_{22}, and off-diagonal entries $b_{12} = b_{21} = -c$. Let $\mathbf{x}$ and λ be as specified by the lemma statement, and assume that $x_2 = 0$. The second element of the vector resulting from multiplying $B\mathbf{x} = \lambda\mathbf{x}$ is $-c \cdot x_1 = \lambda x_2 = 0$. Since $c \neq 0$ by definition (G is a weighted path graph), it must be the case $x_1 = 0$, which implies that $\mathbf{x} = 0$.

For the induction step, assume that the result holds for all $i \leq k$, and consider the standard matrix representation of a weighted path graph on $k+1$ vertices. Let the weight of edge (v_k, v_{k+1}) be c. Let $\mathbf{x}$ and λ be as stated, and assume that $x_{k+1} = 0$. Then $[B\mathbf{x}]_{k+1} = -c \cdot x_k = \lambda x_{k+1} = 0$. Thus $x_k = 0$. Let $\mathbf{x}'$ be the subvector of $\mathbf{x}$ consisting of the first k entries. Note that with $x_{k+1} = 0$ it is the case that $\mathbf{x}'$ and λ meet the lemma conditions for the principle leading minor B_k of B, and that $x'_k = 0$. But B_k is the standard matrix representation for the weighted path graph derived from G by deleting the last edge and vertex. Thus, by the induction hypothesis $\mathbf{x}'$ must be 0; because $x_{k+1} = 0$ this implies that $\mathbf{x} = 0$.

A symmetric argument implies the result for $x_1 = 0$.

Again let B be the standard matrix representation of a weighted path graph G. Let $\mathbf{x}$ be a vector meeting the lemma conditions for λ, and assume that $\mathbf{x}$ has two consecutive zero elements x_i and x_{i+1}. If either $i = 1$ or $i + 1 = n$, $\mathbf{x} = 0$ by the previous argument. Otherwise, $x_{i+1} = 0$ implies that the first i elements of $\mathbf{x}$ and λ meet the lemma conditions for the leading principle minor B_i of B. Note that B_i is the standard matrix representation for some weighted path graph. Thus by the previous result the first i entries of $\mathbf{x}$ are zero. By a symmetric argument for the trailing principle minor, the last $n - i$ entries must also be zero, which gives $\mathbf{x} = 0$.
□

This lemma implies that for eigenvectors of the standard matrix representation of any weighted path graph, neither the first nor the last entry is zero. Likewise, such an eigenvector cannot have two consecutive zero entries. These facts can be used to give a simple proof of the following lemma (for a different proof, see e.g. pp. 910-911 of [28]).

LEMMA 4.5. *All eigenvalues of the standard matrix representation B of a weighted path graph G on n vertices are simple (i.e., have multiplicity one).*

Proof. Let $\mathbf{u}$ and $\mathbf{u}'$ be any two eigenvectors of B for the eigenvalue λ. By Lemma 4.4, $u_n \neq 0$ and $u'_n \neq 0$. Let α be u'_n/u_n; α is non-zero and real. Then $B\,(\alpha\mathbf{u} - \mathbf{u}') = \lambda\,(\alpha\mathbf{u} - \mathbf{u}')$. But the n^{th} element of $(\alpha\mathbf{u} - \mathbf{u}')$ is 0, so by Lemma 4.4, it must be the case that $\alpha\mathbf{u} = \mathbf{u}'$, so $\mathbf{u}$ must be a scalar multiple of $\mathbf{u}'$; it is not a distinct eigenvector.
□

A path graph on n vertices has exactly one automorphism of order two: $\phi(i) = n-i+1$. Thus one can talk about odd and even eigenvectors of a path graph without ambiguity; they are always with respect to this automorphism.

LEMMA 4.6. *Let G be an unweighted path graph on n vertices with Laplacian B. The eigenvector $\mathbf{u}_2$ corresponding to $\lambda_2(B)$ is odd.*

Proof. By Lemma 4.5, $\mathbf{u}_2$ is simple, so by Corollary 4.2, $\mathbf{u}_2$ must be either even or odd. Assume that it is even. We show this leads to a contradiction.

There are two cases to keep track of: n is odd, and n is even. If n is odd, there is a single center vertex $v_{\lceil \frac{n}{2} \rceil}$ (index the vertices along the path from 1 to n). If n is even, there are two center vertices with indices $\frac{n}{2}$ and $\frac{n}{2}+1$; since $\mathbf{u}_2$ is assumed to be even, their entries in $\mathbf{u}_2$ are equal. Thus, by Lemma 4.4, if n is even the eigenvector entries corresponding to the center vertices are non-zero. If n is odd, $\mathbf{u}_2$ is even, and the eigenvector entry for the center vertex is 0, then it is easy to check that changing the signs of all eigenvector entries with index less than the center index gives an odd eigenvector with eigenvalue λ_2, which contradicts the simplicity of λ_2. Thus, the assumption that $\mathbf{u}_2$ is even implies that the eigenvector entries corresponding to the center vertex or vertices must be non-zero. Let this value be c.

Now consider the vector $\mathbf{x} = (-c) \cdot \vec{1} + \mathbf{u}_2$. Recall that $\mathbf{u}_2$ is orthogonal to $\vec{1}$. It is easy to see that $\mathbf{x}$ is even, and since $c \neq 0$,

$$\frac{\mathbf{x}^T B \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \frac{\mathbf{u}_2^T B \mathbf{u}_2}{c^2 n + \mathbf{u}_2^T \mathbf{u}_2} < \frac{\mathbf{u}_2^T B \mathbf{u}_2}{\mathbf{u}_2^T \mathbf{u}_2} = \lambda_2.$$

However, the entries of $\mathbf{x}$ corresponding to the center vertex or vertices are 0, so as above, one can create an odd vector $\mathbf{y}$ such that $\frac{\mathbf{y}^T B \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{\mathbf{x}^T B \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ as follows: set $y_i = x_i,\ i < \frac{n}{2}$ and $y_i = -x_i,\ i > \frac{n}{2}$. Recall the characterization $\lambda_2 = \min_{\mathbf{x} \perp \vec{1}} \frac{\mathbf{x}^T B \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$; $\mathbf{y}$ is orthogonal to $\vec{1}$, so it meets the criteria for the characterization of λ_2, so the assumption that $\mathbf{u}_2$ is even gives $\lambda_2 < \lambda_2$, a contradiction.
□

The reader can easily verify that this theorem also holds for generalized Laplacians (i.e., Fiedler's matrix representation of graphs with positive edge weights) where the automorphism ϕ exists. However, extensions to the standard matrix representation case is not possible because of vertex weights and negative edge weights.

5. A Bad Family of Bounded-Degree Planar Graphs for Spectral Bisection. In this section we present a family of bounded-degree planar graphs that have constant-size separators. However, the separators produced by spectral bisection have size $\Theta(n)$ for both edge and vertex separators.

The family of graphs is parameterized on the positive integers. G_k consists of two path graphs, each on $2k$ vertices, with a set of edges between the two paths as follows: label the vertices of one path from 1 to $2k$ in order (the **upper path**), and label the other path from $2k+1$ to $4k$ in order (the **lower path**). For $1 \leq i \leq k$ there is an edge between vertices $k+i$ and $3k+i$. An example for $k=5$ is shown in Figure 5.1. It is obvious that G_k is planar for any k, and that the maximum degree of any vertex is 3.

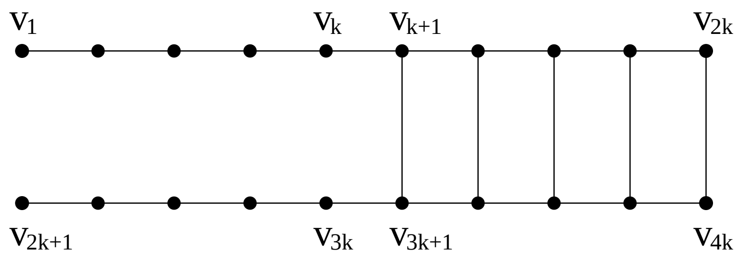

FIG. 5.1. *The Roach Graph for $k=5$*

Note that the graph has the approximate shape of a cockroach, with the section containing edges between the upper and lower paths being the body, and the other sections of the paths being antennae. This terminology allows easy references to parts of the graph.

G_k has one automorphism of order 2 that maps the vertices of the upper path to the vertices of the lower path and vice versa. For the rest of this section, the terms "odd vector" and "even vector" are used with respect to this automorphism. Thus, an even vector $\mathbf{x}$ has $x_i = x_{2k+i}$ for all i in the range $1 \leq i \leq 2k$; an odd vector $\mathbf{y}$ has $y_i = -y_{2k+i}$ for all i, $1 \leq i \leq 2k$.

We can now discuss the structure of the eigenvectors of B_k, the Laplacian of G_k:

LEMMA 5.1. *Any eigenvector $\mathbf{u}_i$ with eigenvalue λ_i of B_k can be expressed as a linear combination of:*

- *an even eigenvector of B_k in which the values associated with the upper path are the same as for the eigenvector with eigenvalue λ_i (if it exists) of a path graph on $2k$ vertices, and*
- *an odd eigenvector of B_k in which the values associated with the upper path are the same as for the eigenvector with eigenvalue λ_i (if it exists) of a weighted graph that consists of a path graph on $2k$ vertices for which the vertex weights of v_{k+1} through v_{2k} have been increased by 2.*

Proof. The fact that we can express any eigenvector of B_k as a sum of odd and even eigenvectors follows by Theorem 4.1 applied with respect to the automorphism of order 2.

The claim about the specific structure of the odd and even eigenvectors of B_k follows from an application of the even-odd decomposition proved in Theorem 4.3, with the odd and even matrix components described in graph form.

□

It is obvious that G_k has a bisector of constant size: cut the edges connecting the antennae to the body. The following theorem shows that spectral bisection gives much larger bisectors for the family of graphs G_k:

THEOREM 5.2. *Spectral bisection produces $\Theta(n)$ edge and vertex separators for G_k for any k.*

Proof. The first step is to show that $\mathbf{u}_2$ is odd. Intuitively, this implies that the spectral method splits the graph into the upper path and the lower path.

Recall that $\lambda_2 = \min_{\mathbf{x}\perp\vec{1}} \frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T\mathbf{x}}$. We construct an odd vector $\mathbf{x}$ such that the quotient $\frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T\mathbf{x}}$ is less than $\frac{\mathbf{y}^T B_k \mathbf{y}}{\mathbf{y}^T\mathbf{y}}$ for any even eigenvector $\mathbf{y}$ orthogonal to $\vec{1}$ ($\vec{1}$ is the smallest even eigenvector). This requires a proof that $\frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T\mathbf{x}}$ is less than the second smallest even eigenvalue. From Lemma 5.1 above, the second smallest even eigenvalue of B_k is the same as the second smallest eigenvalue μ_2 of the Laplacian B of a path graph G on $2k$ vertices; it is well-known that $\mu_2 = 4\sin^2(\frac{\pi}{4k})$ (see for example [21]).

Let $\mathbf{z}$ be the eigenvector of B corresponding to μ_2. Construct $\mathbf{x}$ as follows:

$$x_i = \begin{cases} z_i & 1 \le i \le k, \\ z_{4k-i+1} & 2k+1 \le i \le 3k, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

That is, assign the first k values from the path G to the upper antenna of the roach, working in the direction towards the body, and assign the last k entries from G to the lower antenna, working from the body outward. Since $\mathbf{z}$ and $\mathbf{x}$ have the same set of non-zero entries, $\mathbf{x}^T\mathbf{x} = \mathbf{z}^T\mathbf{z}$. Likewise, since $\mathbf{z}$ is orthogonal to the "all-ones" vector, so is $\mathbf{x}$.

To see that $\mathbf{x}^T B_k \mathbf{x} < \mathbf{z}^T B \mathbf{z}$, recall (3.1) from Section 3: for Laplacian B and vector $\mathbf{y}$,

$$\mathbf{y}^T B \mathbf{y} = \sum_{(v_i,v_j)\in E} (y_i - y_j)^2.$$

For every edge in G except one, there is an edge in G_k that contributes the same value to this sum. The one exception is the edge (v_k, v_{k+1}) in G. Since $\mathbf{z}$ is an odd vector by Lemma 4.6, and since $\mathbf{z}$ has an even number of entries, $z_k = -z_{k+1}$. By Lemma 4.4, it is not possible for both z_k and z_{k+1} to be zero, so z_k is equal to some non-zero value c, and this edge contributes $4c^2$ to the value of $\mathbf{z}^T B\mathbf{z}$. On the other hand, there are two edges in G_k that contribute non-zero values and that do not have corresponding edges in G: (v_k, v_{k+1}) and (v_{3k}, v_{3k+1}). Each of these edges contributes c^2 to $\mathbf{x}^T B_k \mathbf{x}$. Thus

$$\mathbf{x}^T B_k \mathbf{x} = \mathbf{z}^T B \mathbf{z} - 4c^2 + 2c^2 < \mathbf{z}^T B \mathbf{z}.$$

Since $\mathbf{x}^T\mathbf{x} = \mathbf{z}^T\mathbf{z}$,

$$\lambda_2(G_k) \le \frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T\mathbf{x}} < \frac{\mathbf{z}^T B \mathbf{z}}{\mathbf{z}^T\mathbf{z}} = 4\sin^2(\frac{\pi}{4k}).$$

That is, the second smallest eigenvalue of B_k is less than any non-zero even eigenvalue, and is thus odd by Theorem 4.1.

We still need to show that there are not too many zero entries in $\mathbf{u}_2$ (spectral bisection as defined in this paper does not separate vertices with the same value). Since $\mathbf{u}_2$ is an odd vector and since the odd component of G_k is a weighted path graph, Lemmas 4.4 and 5.1 imply that $\mathbf{u}_2$ cannot have consecutive zeros, and the values corresponding to vertices $2k$ and $4k$ are non-zero. Thus the edge separator generated by spectral bisection must cut at least half the edges between the upper and lower

paths; since none of these edges share an endpoint, the cover used in generating the vertex separator must include at least this number of vertices.
□

Recently Spielman and Teng have presented an algorithm that recursively applies a spectral separator method to give bisections of planar graphs guaranteed to be $O(\sqrt(n))$; their technique applied to the roach graph gives a bisection of constant size. See Section 8 for details.

6. A Bad Family of Graphs for the "Best Threshold Cut" Algorithm. While the roach graph defeats spectral bisection, the second smallest eigenvector can still be used to find a small separator using the "best threshold cut" algorithm. In particular, Theorem 3.1 implies that considering all threshold cuts induced by $\mathbf{u}_2$ produces a constant-size cut: If q_{min} is the minimum cut quotient for these cuts, then

$$q_{min} \leq \sqrt{\lambda_2(2\Delta - \lambda_2)} \leq \frac{\sqrt{6}\pi}{4k},$$

which implies q_{min} is $O(\frac{1}{n})$. Since the denominator of q_{min} is less than or equal to $\frac{n}{2}$, the number of edges in this cut must be bounded by a constant.

In this section we show that there is a family of graphs for which the "best threshold cut" algorithm does poorly. The graphs in this family consist of crossproducts of path graphs and double trees. A **double tree** is two complete binary trees of k levels for some $k > 0$ connected by an edge between their respective roots.

The following two bounds are proved in [16]:

LEMMA 6.1. *For a complete balanced binary tree on $k \geq 3$ levels and $n = 2^k - 1$ vertices, $\frac{1}{n} < \lambda_2 < \frac{2}{n}$.*

For double trees where each of the component trees has k levels, $n = 2^{k+1} - 2$. The following bound applies:

LEMMA 6.2. *For a double tree on $n \geq 14$ vertices, $\frac{1}{n} < \lambda_2 < \frac{4}{n}$.*

The **tree-cross-path graph** consists of the crossproduct of a double tree on p_1 vertices and a path graph on p_2 vertices. We show that there are tree-cross-path graphs that defeat the "best threshold cut" algorithm.

We formally state the result for this section as follows:

THEOREM 6.3. *There exists a graph G for which the "best threshold cut" algorithm finds a separator S such that the cut quotient for S is bigger than $i(G)$ by a factor as large (to within a constant) as allowed by the bounds from Theorem 3.1.*

Proof. Let G be the tree-cross-path graph that is the crossproduct of a double tree of size p and a path of length $cp^{\frac{1}{2}}$ for some c in the range $3.5 \leq c < 4$. To insure that the double tree and the path have integer sizes, restrict p to integers of the form $2^k - 2$ for $k \geq 4$. Then choose c in the range specified such that $cp^{\frac{1}{2}}$ is an integer (the choice of p insures there is an integer in this range).

Recall that the eigenvalues of a graph crossproduct are all pairwise sums of the eigenvalues from the graphs used in the crossproduct operation. Let ν_2 be the second smallest eigenvalue of the double tree on p vertices, and let μ_2 be the second smallest eigenvalue for the path on $cp^{\frac{1}{2}}$ vertices. If $\mu_2 < \nu_2$, then λ_2 for the crossproduct is μ_2 (i.e., μ_2 added to the zero eigenvalue of the double tree). Since $\mu_2 = 4\sin^2\left(\frac{\pi}{2cp^{\frac{1}{2}}}\right)$ and $\nu_2 \geq \frac{1}{p}$ (by Lemma 6.2 and the choice of p), it is necessary to show that $4\sin^2\left(\frac{\pi}{2cp^{\frac{1}{2}}}\right) < \frac{1}{p}$. Reorganizing, simplifying, and noting that $\sin(\theta) < \theta$

for $0 < \theta \leq \frac{\pi}{2}$, it is sufficient to show that $\pi < c$. Clearly by the choice of c this inequality holds.

Note that the tree-cross-path graph can be thought of as $cp^{\frac{1}{2}}$ copies of the double tree, each corresponding to one vertex of the path graph. Each vertex in the i^{th} copy of the double tree is connected by an edge to the corresponding vertex in copies $i-1$ and $i+1$. This description allows one to construct the eigenvector for the second smallest tree-cross-path eigenvalue as follows: Assign each vertex in double tree copy i the value for vertex i in the path graph eigenvector for μ_2. Note that this is the only possible eigenvector since path graph eigenvalues are simple by Lemma 4.5.

Now consider any copy of the double tree: every vertex in that copy gets the same value in the characteristic valuation. Thus the cut S made by the "best threshold cut" algorithm must separate at least two copies of the double tree, and thus must cut at least p edges. There is a bisection S^* of size $cp^{\frac{1}{2}}$ (cut the edge between the roots in each double tree); because this cut is a bisection, the ratio between the cut quotient q for S and $i(G)$ is at least as large as the ratio between the sizes of these cuts:

$$\frac{q}{i(G)} \geq \frac{|S|}{|S^*|} \geq \frac{p}{cp^{\frac{1}{2}}} = \Omega\left(p^{\frac{1}{2}}\right).$$

From Theorem 3.1,

$$\frac{\lambda_2}{2} \leq i(G) \leq q \leq \sqrt{\lambda_2(2\Delta - \lambda_2)}.$$

This plus the fact that the tree-cross-path graph has bounded degree ($\Delta = 5$) implies that

$$\frac{q}{i(G)} \leq \frac{2\sqrt{\lambda_2(2\Delta - \lambda_2)}}{\lambda_2} = O\left(\frac{1}{\sqrt{\lambda_2}}\right) = O\left(p^{\frac{1}{2}}\right).$$

These two bounds imply that, to within a constant factor, the ratio is as large as possible, and the theorem holds.
□

7. A Bad Family of Graphs for Generalized Spectral Algorithms.

7.1. Purely Spectral Algorithms. In Section 2 we noted that many variations of spectral partitioning have been suggested. In this section we extend the results of the previous section to cover those variations and many other possibilities, including algorithms that use some number k (where k might depend on n) of the eigenvectors corresponding to the k smallest non-zero eigenvalues. In particular, consider algorithms that meet the following restrictions:

- The algorithm computes a value for each vertex using only the eigenvector components for that vertex from k eigenvectors corresponding to the smallest non-zero eigenvalues (for convenience, We refer to these as the k **smallest eigenvectors**). The function computed can be arbitrary as long as its output depends only on these inputs.
- The algorithm partitions the graph by choosing some threshold t and then putting all vertices with values greater than t on one side of the partition, and the rest of the vertices on the other side.
- The algorithm is free to compute the break point t in any way; e.g., checking the cut quotient for all possible breaks and choosing the best one is allowed.

We call such an algorithm **purely spectral**.

7.2. Purely Spectral Algorithms that Use a Constant Number of Eigenvectors. The following theorem gives a bound on how well such algorithms do when the number of eigenvectors used is a constant:

THEOREM 7.1. *Consider the purely spectral algorithms that use the k smallest eigenvectors for k a fixed constant. Then there exists a family of graphs $\mathcal{G}$ such that $G \in \mathcal{G}$ has a bisection S^* with $|S^*| \geq (k^2 n)^{\frac{1}{3}}$, and such that any purely spectral algorithm using the k smallest eigenvectors produces a separator S for G such that $|S| \geq \left(\frac{|S^*|}{\pi k+1}\right)^2$.*

Proof. We show that $\mathcal{G}$ is the set of tree-cross-path graphs that are the crossproducts of double trees of size p (where p is an integer of the form $2^j - 2$ for some $j \geq 4$) and paths of length $cp^{\frac{1}{2}}$, where c is a constant chosen such that $\pi k < c \leq \pi k + 1$ and $cp^{\frac{1}{2}}$ is an integer.

Using slight modifications of arguments from Theorem 6.3, one can show the following: For graphs in $\mathcal{G}$, the k smallest positive path eigenvalues are less than the smallest positive eigenvalue of the double tree. This implies that every vertex in a particular copy of the double tree receives the same set of values from the k eigenvectors. Thus the purely spectral algorithm assigns the same value to each vertex in that copy. This implies that S, the separator produced, must separate at least two copies of the double tree, and thus must cut at least p edges.

There is a bisection S^* of size $cp^{\frac{1}{2}}$ (it cuts the edge between the roots in each double tree); because $n = cp^{\frac{3}{2}}$ and $c > k$, it is the case that $|S^*| > k^{\frac{2}{3}} n^{\frac{1}{3}}$. It is obvious that

$$|S| \geq \left(\frac{|S^*|}{c}\right)^2;$$

since $c \leq \pi k + 1$, the theorem holds.
□

Note that for the case in which k is constant, the following results apply for the family of graphs described in the preceding theorem:

- the cut quotient q_S is no better than the best cut quotient q_{min} produced by considering all threshold cuts for $\mathbf{u}_2$, and
- the gap between $i(G)$ and q_{min} is as large as possible (within a constant factor) with respect to Theorem 3.1. The bound on $|S^*|$ implies that the spectral separator is bigger by a factor of at least a constant times $n^{\frac{1}{3}}$.

These results can be shown using techniques from the previous section. Thus, for such graphs, using k eigenvectors does not improve the performance of the "best threshold cut" algorithm.

These results also hold for certain variants of the definition of "purely spectral". For example, Chan, Gilbert, and Teng have proposed using the entries of eigenvectors 2 through $d + 1$ of the Laplacian as spatial coordinates for the corresponding vertices of a graph [7]. The graph is then partitioned using a geometric separator algorithm [20],[15]. If this technique is applied (using a fixed d) to the counterexample graph used in the proof above, all vertices in a particular copy of the double tree end up with the same coordinates; the geometric algorithm then cuts between copies of the double tree, yielding the same bad cuts as in the proof.

7.3. Purely Spectral Algorithms that Use More than a Constant Number of Eigenvectors. There are still a number of open questions about the performance of purely spectral algorithms that use more than a constant number of eigenvectors (in particular, how well can such algorithms do if they use all the eigenvectors?). However, just using more than a constant number of eigenvectors is not sufficient to guarantee good separators. In particular, the counterexamples and arguments in the previous sections can be extended to prove the following theorem:

THEOREM 7.2. *For sufficiently large n and $0 < \epsilon < \frac{1}{4}$, there exists a bounded-degree graph G on n vertices such that any purely spectral algorithm using the n^ϵ smallest eigenvectors produces a separator S for G with a cut quotient greater than $i(G)$ by at least a factor of $n^{\left(\frac{1}{4}-\epsilon\right)} - 1$.*

Proof. Once again, let G be the tree-cross-path graph. As in the previous two proofs, choose p_1 (the double-tree size) and p_2 (the path size) such that the smallest n^ϵ eigenvalues of the crossproduct are the same as the smallest n^ϵ eigenvalues of the path graph. Once again, a purely spectral algorithm separates two adjacent double trees, while the edges between the roots of the double trees form a better separator. It remains to choose p_1 and p_2 such that the claim about the smallest eigenvalues of the crossproduct holds, and to show that the resulting cut is bad.

Set p_1 to some arbitrary positive integer p, subject to the conditions presented below to insure that p is sufficiently large. Then set $p_2 = \left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)} \right\rceil$. Note that p can be chosen sufficiently large such that

$$p > p^{\left(\frac{1}{2}+2\epsilon\right)} + 1 > p_2.$$

This implies that $p > n^{\frac{1}{2}}$, where $n = p_1 p_2$. Note that this allows one to show easily that $n^\epsilon < p^{2\epsilon} < p_2$ (i.e., since the algorithm uses n^ϵ eigenvectors, this argument requires the path graph to have at least that many eigenvalues, and thus be at least that long). Also note that even for fairly small p, $p_2 < 2p^{\left(\frac{1}{2}+2\epsilon\right)}$, which implies that

$$n < 2p^{\left(\frac{3}{2}+2\epsilon\right)}. \tag{7.1}$$

Now consider the ratio of the size of the cut produced by cutting the double-tree edges to the size of the cut produced by a purely spectral method under the assumption that the n^ϵ smallest eigenvalues are the same as for the path graph. As in previous proofs, this ratio is at least as large as the ratio between the number of edges cut. Thus, for sufficiently large p, the ratio is at least

$$\frac{p}{\left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)} \right\rceil} > p^{\left(\frac{1}{2}-2\epsilon\right)} - 1 > n^{\left(\frac{1}{4}-\epsilon\right)} - 1.$$

All that is left to prove is the assumption about the smallest eigenvalues. If $\alpha = \frac{1}{2} - 2\epsilon$, then $\alpha > 0$ and inequality (7.1) above can be written as

$$n < 2p^{(2-\alpha)}. \tag{7.2}$$

Recall that the eigenvalues of a path graph on k vertices are $4\sin^2(\frac{\pi i}{2k})$ for $0 \le i < k$, and that λ_2 for a double tree on $p \ge 14$ vertices is greater than or equal to $\frac{1}{p}$. It remains to show that for p sufficiently large,

$$4\sin^2\left(\frac{\pi n^\epsilon}{2\left\lceil p^{\left(\frac{1}{2}+2\epsilon\right)} \right\rceil}\right) < \frac{1}{p}.$$

Reorganizing, simplifying, noting that $\sin(\theta) < \theta$ for $0 < \theta \leq \frac{\pi}{2}$, and applying inequality (7.2) above, it is sufficient to show that there is a sufficiently large p such that

$$\frac{\pi\left(2p^{(2-\alpha)}\right)^{\epsilon}}{2p^{\left(\frac{1}{2}+2\epsilon\right)}} < \frac{1}{2p^{\frac{1}{2}}}, \text{ or } \pi 2^{\epsilon} < p^{\alpha\epsilon}.$$

Clearly this inequality holds for sufficiently large p.
□

8. A Note on More Recent Developments. Subsequent to the initial appearance of these results [17], Spielman and Teng published a paper on the performance of spectral partitioning algorithms [27]. Their work has several parts, including:

- A proof that for any bounded-degree planar graph, $\lambda_2 = O(n^{-1})$, and that for well-shaped meshes in d dimensions, $\lambda_2 = O(n^{-\frac{2}{d}})$.
- A new proof of a theorem credited to Mihail that extends bounds on quotient cuts to all vectors with small Rayleigh quotients.
- A recursive spectral bisection algorithm. The algorithm produces $O(n^{\frac{1}{2}})$ bisectors for planar graphs and $O(n^{1-\frac{1}{d}})$ bisectors for well-shaped d-dimensional meshes.
- A new bounded-degree planar counterexample graph for which "best threshold cut" gives a poorly-balanced separator.

It is interesting to consider how those results relate to the results in this paper.

We have shown that there are bounded-degree planar graphs for which spectral bisection based on $\mathbf{u}_2$ alone gives a cut of size $\Theta(n)$. Spielman and Teng's recursive spectral bisection algorithm, however, produces constant size bisections for our counterexamples. Thus their algorithm gives a greatly improved, if somewhat more expensive, result. Their bounded-degree counterexample graph is an interesting advance over the roach graph in that it gives a bounded-degree planar graph with both a bad bisection and a poorly-balanced "best threshold cut".

As for the tree-cross-path examples, the two papers illustrate the difference between guarantees on the size of a balanced cut versus its optimality. If on the first cut, the recursive algorithm produces a bisection that is large relative to the best bisection, the recursion will not improve the bisection. (This is the case for the tree-cross-path graph.) Examples exist for well-shaped meshes. The following graph was suggested by John Gilbert: let a double grid be a pair of $k \times k$ square grids that share a single common corner. As shown in [19], λ_2 of the double grid is $\Theta(\frac{1}{k^2 \log k})$. The double-grid-cross-path graph is a crossproduct between a double grid graph and a path graph. Note that for a suitable constant c, if the path has length $ck\sqrt{\log k}$, the path graph contributes the second smallest eigenvalue of the double-grid-cross-path. Following an analysis similar to that in Theorem 6.3, one can show that the "best threshold cut" for such a double-grid-cross-path is a bisection of size $\Theta(k^2)$ that splits the graph between two copies of the double grid. It is easy to check that the recursive algorithm also returns this cut, and thus does not improve the quality of the single spectral cut. However, this example has a bisection of size $\Theta(k\sqrt{\log k})$ (separate the grids at their common points). The larger bisection meets the guarantee for three-dimensional grids (n here is $ck^3\sqrt{\log k}$), but is not optimum.

Acknowledgments. Special thanks are due Dafna Talmor and Doug Tygar, who provided valuable comments on earlier versions of this paper.

REFERENCES

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

[2] N. Alon, Z. Galil, and V. D. Milman, *Better expanders and superconcentrators*, Journal of Algorithms, 8 (1987), pp. 337–347.

[3] N. Alon and V. D. Milman, λ_1*, isoperimetric inequalities for graphs, and superconcentrators*, Journal of Combinatorial Theory, Series B, 38 (1985), pp. 73–88.

[4] B. Aspvall and J. R. Gilbert, *Graph coloring using eigenvalue decomposition*, SIAM Journal on Algebraic and Discrete Methods, 5 (1984), pp. 526–538.

[5] E. R. Barnes, *An algorithm for partitioning the nodes of a graph*, SIAM Journal on Algebraic and Discrete Methods, 3 (1982), pp. 541–550.

[6] R. Boppana, *Eigenvalues and graph bisection: An average-case analysis*, in 28th Annual Symposium on Foundations of Computer Science, Los Angeles, October 1987, IEEE, pp. 280–285.

[7] T. F. Chan, J. R. Gilbert, and S. H. Teng, *Geometric spectral partitioning*, Tech. Report CSL-94-15, Xerox PARC, July 1994. Revised January 1995.

[8] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs*, Academic Press, New York, 1979.

[9] W. E. Donath and A. J. Hoffman, *Lower bounds for the partitioning of graphs*, IBM Journal of Research and Development, 17 (1973), pp. 420–425.

[10] J. Falkner, F. Rendl, and H. Wolkowicz, *A computational study of graph partitioning*, Mathematical Programming, 66 (1994), pp. 211–240.

[11] M. Fiedler, *Algebraic connectivity of graphs*, Czechoslovak Mathematical Journal, 23 (1973), pp. 298–305.

[12] ———, *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czechoslovak Mathematical Journal, 25 (1975), pp. 619–633.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness*, Freeman, San Francisco, 1979.

[14] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified NP-Complete graph problems*, Theoretical Computer Science, 1 (1976), pp. 237–267.

[15] J. Gilbert, G. Miller, and S. H. Teng, *Geometric mesh partitioning: Implementation and experiments*, in 9th International Parallel Processing Symposium, Santa Barbara, April 1995, IEEE.

[16] S. Guattery and G. Miller, *On the performance of spectral graph partitioning methods*, Tech. Report CMU-CS-94-228, Carnegie Mellon University, December 1994.

[17] ———, *On the performance of spectral graph partitioning methods*, in 6th ACM-SIAM Symposium on Discrete Algorithms, San Francisco, January 1995, ACM-SIAM, pp. 233–242.

[18] L. Hagen and A. B. Kahng, *New spectral methods for ratio cut partitioning and clustering*, IEEE Transactions on Computer-Aided Design, 11 (1992), pp. 1074–1085.

[19] N. Kahale, *A semidefinite bound for mixing rates of Markov chains*, in Proc. Fifth Integer Programming and Combinatorial Optimization Conference, 1996, pp. 190–203. Lecture Notes in Computer Science 1084.

[20] G. L. Miller, S. H. Teng, and S. A. Vavasis, *A unified geometric approach to graph separators*, in 32nd Annual Symposium on Foundations of Computer Science, Puerto Rico, Oct 1991, IEEE, pp. 538–547.

[21] B. Mohar, *The Laplacian spectrum of graphs*, in Sixth International Conference on the Theory and Applications of Graphs, 1988, pp. 871–898.

[22] ———, *Isoperimetric numbers of graphs*, Journal of Combinatorial Theory, Series B, 47 (1989), pp. 274–291.

[23] A. Pothen, H. D. Simon, and K. P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM Journal on Matrix Analysis and Applications, 11 (1990), pp. 430–452.

[24] F. Rendl and H. Wolkowicz, *A projection technique for partitioning the nodes of a graph*, Annals of Operations Research, 58 (1995), pp. 155–180.

[25] H. D. Simon, *Partitioning of unstructured problems for parallel processing*, Computing Systems in Engineering, 2 (1991), pp. 135–148.

[26] A. Sinclair and M. Jerrum, *Approximate counting, uniform generation and rapidly mixing Markov chains*, Information and Computation, 82 (1989), pp. 93–133.

[27] D. A. Spielman and S. H. Teng, *Spectral partitioning works: Planar graphs and finite element meshes*, Tech. Report UCB CSD-96-898, U.C. Berkeley, 1996. An extended abstract appeared in the 37th Annual Symposium on Foundations of Computer Science.

[28] D. M. Young and R. T. Gregory, *A Survey of Numerical Mathematics*, vol. II of Addison-Wesley Series in Mathematics, Addison-Wesley, Reading, MA, 1973.