On the Performance of Spectral Graph Partitioning Methods *†

Stephen Guattery Gary L. Miller

Abstract

Computing graph separators is an important step in many graph algorithms. A popular technique for computing graph separators involves spectral methods. However, there is not much theoretical analysis of the quality of the separators produced by spectral methods; instead it is usually claimed that such 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 version of the spectral method that allows the use of some specified number of the eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix of a graph; for such algorithms, we show that if they use a constant number of eigenvectors, then there are graphs for which they do no better than using only the second smallest eigenvector. We also show that in this case the algorithm based on the second smallest eigenvector performs poorly with respect to theoretical bounds. Even if an algorithm meeting our generalized definition uses up to 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.

1 Introduction

Spectral methods (i.e., methods using the eigenvalues and eigenvectors of a matrix representation of a graph) are widely used to compute graph separators. Typically, the Laplacian matrix[1] representation B of a graph G is used. 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 $\mathbf{u}_2$ [PSL90, HK92]. 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.

Although spectral methods are popular, there is little theoretical analysis of how well they do in producing small separators. Instead, it is usually claimed that such methods "work well in practice," and tables of results for specific examples are often included in papers [PSL90]. 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 could be from optimal. As 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 counterexamples for which each of the algorithms provide 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 is a popular technique that consists of bisecting a graph by partitioning the vertices in to two equal-sized sets based on each vertex's entry in the eigenvector corresponding to the second-smallest eigenvalue. A graph in our first counterexample class looks like a ladder with the top 2/3 of its rungs kicked out (see Figure 1); a straightforward spectral bisection algorithm cuts the remaining rungs, where the optimal bisection is made by cutting across the ladder above the remaining rungs. (We refer to this graph as the "roach graph" because its outline looks roughly like the body of a cockroach and two long antennae.)

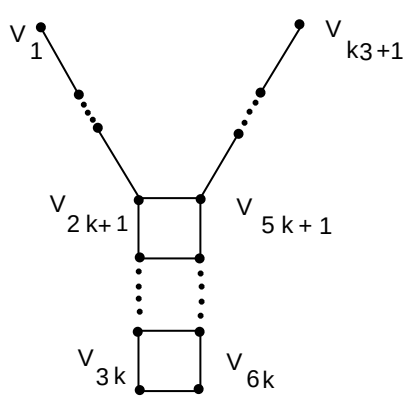

Figure 1: The Roach Graph

*School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213-3890.

†This work was supported in part by NSF Grant CCR-9016641.

[1] The Laplacian B of a graph G on n vertices is an $n \times n$ 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.

It is possible to modify the simple spectral algorithm in ways that allow it to generate a better separator for the roach graph. Some examples of possible modifications are presented in [HK92]; these examples still use a partition based on $\mathbf{u}_2$. We consider a modified simple spectral algorithm based on the most general of these suggested modifications, and present a second counterexample class of graphs that defeats this algorithm. This class of graphs consists of crossproducts of path graphs and graphs consisting of a pair of complete binary trees connected by an edge between their roots.

Finally, we consider a more general definition of purely spectral separator algorithms that subsumes the two preceding algorithms. This general definition allows the use of some specified number of eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix a graph. For any such algorithm, we show that if it uses a fixed number of eigenvectors, then there are graphs for which it does no better than using the modified simple spectral algorithm. Further, the separator produced by the modified simple spectral algorithm is as bad as possible (to within a constant) with respect to theoretical bounds on 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 $n^{\frac{1}{4}-\epsilon} - 1$ of the isoperimetric number. We also note that graphs in this class used to construct these counterexamples can be extended to graphs that could conceivably be used as three-dimensional finite-element meshes – that is, graphs that could be encountered in practice.

This paper makes one additional contribution: Our counterexamples have simple structures and intuitively would 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 that exist in our counterexamples.

Specifics are given in the text that follows. Section 2 gives some history of spectral methods and details of the algorithms we discuss in this paper. Section 3 gives the graph and matrix terminology that we will use, and presents some useful facts about Laplacian matrices. Section 4 gives the counterexample for the simple spectral algorithm; Section 5 gives the counterexample for the modified simple spectral method. Finally, Section 6 discusses our generalized notion of spectral separator algorithms, and show that there are graphs for which any such algorithm performs poorly.

2 Spectral Methods for Computing Separators

The roots of spectral partitioning go back to Hoffmann and Donath [DH73], who proved a lower bound on the size of the minimum bisection of a graph, and Fiedler [Fie73][Fie75], who explored the properties of λ_2 and its associated eigenvector for the Laplacian of a graph. There has been much subsequent work, including Barnes's partitioning algorithm [Bar82], Boppana's work that included a stronger lower bound on the minimum bisection size [Bop87], and the particular bisection and graph partitioning problems that we are considering in this paper [HK92] [PSL90] [Sim91]. (We note that spectral methods have not been limited to graph partitioning; work has been done using the spectrum of the adjacency matrix in graph coloring [AG84] and using the Laplacian spectrum to prove theorems about expander graph and superconcentrator properties [AM85] [Alo86] [AGM87]. 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 [Moh89]. Reference [CDS79] 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 [PSL90]. We will refer to this algorithm as the simple spectral method. The simple spectral method 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 element form one part of the partition; the rest of the vertices form the other part. The set of all edges between the two parts forms an edge separator.
- If a vertex separator is desired, it can be computed from the edge separator as described in the next section.

The simple spectral method gives a bisection and, since the graph bisection problem is NP-complete [GJ79], may not give an optimum result. That is, the simple spectral algorithm is a heuristic method. A number of modifications have been proposed that may improve on its performance. The following modifications, all of which use the characteristic valuation, are presented in [HK92][2]:

[2] These revised heuristics can give different splits than bisec-

- 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 $1 \leq i \leq n-1$, consider the ratio for the separator produced by splitting the vertices into those with sorted index $\leq i$ and those with sorted index $> i$. Choose the split that provides the best separator ratio.

Note that the last idea subsumes the first two. We will consider a variant of this algorithm below. Since this algorithm does not consider the case where multiple vertices may have the same value, we will specify that only splits between vertices with different values are considered (we refer to such cuts as **threshold cuts**). We call this algorithm the **modified simple spectral algorithm**; the fact that we've slightly changed 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 the partitions generated by the set of eigenvectors for some number of smallest eigenvalues: for each vertex, a value would be assigned by computing a function of that vertex's eigenvector components. Partitions could then be 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 the simple spectral method may produce a bad separator for a bounded-degree planar graph in Section 4; first, however, we need to introduce some terminology and background results.

tions. In such cases, we will use the separator ratio or cut quotient (defined in Section 3) in judging how close the split is to optimal. Computing the separator with the minimum ratio is NP-hard (see, e.g., [LR88]).

3 Terminology, Notation, and Background Results

We will 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 will 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 $|G|$ will be sometimes be used as a shorthand for $|V(G)|$. When it is clear what graph we are referring to, we will use n to denote $|V|$.

We will use capital letters to represent matrices and bold lower-case letters for vectors. For a matrix A, a_{ij} or $[A]_{ij}$ will represent the element in row i and column j; for the vector $\mathbf{x}$, x_i or $[\mathbf{x}]_i$ will represent the i^{th} entry in the vector. The notation with square brackets is useful in cases where adding subscripts to lower-case letters would be awkward (e.g., where the matrix or vector name is already subscripted). 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, we will index the eigenvalues of an $n \times n$ matrix in non-decreasing order. λ_1 will represent the smallest eigenvalue, and λ_n the largest. For $1 < i < n$, we have $\lambda_{i-1} \leq \lambda_i \leq \lambda_{i+1}$. We will use $\mathbf{u}_i$ to represent the eigenvector corresponding to λ_i.

We will use the term **path graph** for a tree that has exactly two vertices of degree one. That is, a path graph is a graph consisting of exactly a path.

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 $(u,u') \in E(G)$.

For a connected graph G, an **edge separator** is a set S of edges such that if removed would break the graph into two (not necessarily connected) components G_1 and G_2 such that there are no edges between G_1 and G_2. (We will assume that an edge separator is a minimal set with respect to the particular 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 such that there are no edges between G_1 and G_2 (again, we'll assume that such a separator is minimal). The goal in finding separators is to find a small separator that breaks the graph into two fairly large pieces; often this notion of "large pieces" is expressed as a restriction

that the number of vertices in either component be at least some specified fraction of the number of vertices in G. For edge separators, this can be stated more generally in terms of the **separator ratio** ρ, defined as $|S|/(|G_1| \cdot |G_2|)$. The **optimum ratio separator** ρ_{opt} is the one that minimizes the separator ratio over all separators for a particular graph [LR88].

A related concept (again, for edge separators) is the **isoperimetric number** $i(G)$, defined as:

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

We will refer to the quantity $|S|/\min(|G_1|, |G_2|)$ as the **cut quotient** for the edge separator S. It is easy to see that $n\rho_{opt} > i(G) \geq n\rho_{opt}/2$.

It is well known that an edge separator S can easily be converted into a vertex separator S' by considering the bipartite graph induced by S (where the parts of the bipartition are determined by the components G_1 and G_2), and setting S' to be a minimum edge cover for that graph.

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. (For such representations we will assume that the vertices have been numbered to correspond to the indices used in the matrix.)

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 as defined above. Then the Laplacian representation of G is the matrix $B = D - A$.

The following are some useful facts about the Laplacian matrix:

- The Laplacian is symmetric positive semidefinite, so all its eigenvalues are greater than or equal to 0 (see e.g. [Moh88]).

- A graph G is connected if and only if 0 is a simple eigenvalue of G's Laplacian (see e.g. [Moh88]).

- If G is a crossproduct of two graphs G_1 and G_2, then the eigenvalues of G's Laplacian are all pairwise sums of the eigenvalues of G_1 and G_2 (see e.g. [Moh88]).

- For any vector $\mathbf{x}$ and Laplacian matrix B of the graph G, we have (see e.g. [HK92]):

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

- For a graph G and its Laplacian B, λ_2 of B figures in upper and lower bounds on the isoperimetric number for G [Moh89]. In particular, we have:

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

where Δ is the maximum degree of any vertex in G. This implies the size of any bisection is at least $\frac{n\lambda_2}{4}$.

The proof of the preceding upper bound 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 will 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 $> t_i$, and those with values $\leq t_i$. 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 [Moh89]:

THEOREM 3.1. *Let G be a connected graph with maximal vertex degree Δ with second smallest eigenvalue λ_2. Further, let G not be any of K_1, K_2, or K_3. Let q_{min} be as defined above. Then*

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

This can be extended to the separator ratio for the best $\mathbf{u}_2$ cut in the obvious way.

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) (we consider a zero edge weight to indicate the lack of an edge). The **generalized Laplacian** B for a weighted graph G 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 Laplacian matrix of a graph is also a generalized Laplacian where the vertex weights are taken to be the degrees of the vertices, and all edge weights are 1.

Note that any generalized Laplacian is a real symmetric matrix, so any theorems for such a matrix will apply. The following two theorems about the interlacing of eigenvalues are particularly useful properties of the generalized Laplacian.

The **First Interlacing Property** (the following statement is from page 411 of reference [GL89]; the proof is in [Wil65]): If A_r denotes the leading $r \times r$ principal submatrix of an $n \times n$ symmetric matrix A, then for $r = 1 : n-1$ the following interlacing property holds:

$$\lambda_{r+1}(A_{r+1}) \geq \lambda_r(A_r) \geq \lambda_r(A_{r+1}) \geq \ldots$$
$$\ldots \geq \lambda_2(A_{r+1}) \geq \lambda_1(A_r) \geq \lambda_1(A_{r+1}).$$

The **Second Interlacing Property** (the following statement is a restricted version of a theorem from page 412 of reference [GL89]; the proof is in [Wil65]): Suppose $B = A + \alpha \mathbf{c}\mathbf{c}^T$, where A is a real symmetric $n \times n$ matrix, $\mathbf{c}$ is a real vector of length 1, and α is real and greater than 0. Then for all i, $1 \leq i \leq n-1$ we have:

$$\lambda_i(A) \leq \lambda_i(B) \leq \lambda_{i+1}(A)$$

The Second Interlacing Property implies that if an edge e is added to a graph G to produce the graph G', then $\lambda_2(G) \leq \lambda_2(G')$.

3.1 The Structure of Laplacian and Generalized Laplacian Eigenvectors

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 first set concern eigenvalues of Laplacian matrices 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 on the vertices.

For weighted graphs, we add two conditions to the definition of automorphism: 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 Laplacian matrices for graphs under the standard definition of automorphism, and for generalized Laplacians for weighted graphs under the generalized definition of automorphisms for weighted graphs.

Let G be a graph with an automorphism ϕ of order 2. Let B be the Laplacian of G. 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) that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal.

THEOREM 3.2. **[Even-Odd Eigenvector Theorem]** *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 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 generalized Laplacian of G, and ϕ a weighted graph automorphism of order 2.*

The proofs of this and the other results in this section are given in the complete version of this paper.

COROLLARY 3.1. *Let B be the (generalized) Laplacian 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.*

THEOREM 3.3. **[Even-Odd Graph Decomposition Theorem]** *For every (weighted) graph G with (generalized) Laplacian B and an automorphism ϕ of order 2, there exist two smaller weighted graphs G_{odd} and G_{even} such that the eigenvalues of B_{odd} (the generalized Laplacian of G_{odd}) are the same as the odd eigenvalues of B, the eigenvalues of B_{even} (the generalized Laplacian of G_{even}) are the same as the even eigenvalues of B, and $|V(G_{odd})| + |V(G_{even})| = |V(G)|$. Further, a complete set of eigenvectors of B can be constructed from the eigenvectors of B_{odd} and B_{even} in a straightforward way.*

Though the proof of this theorem is too long to include here, we can give the construction of G_{odd} and G_{even} and the construction of the eigenvectors of B from those of B_{odd} and B_{even}. First we need 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 will call a subset of V_m that consists of exactly one vertex from each such orbit a **representative set** and denote it V_r.

G_{odd} is a weighted graph on V_r. Start with the subgraph of G induced by V_r, with the weight of vertex v_i equal to the weight of v_i in G and the weight of edge (v_i, v_j) equal to its weight in G. Adjust the weights according to the following two rules:

- **Vertex weight adjustment rule:** For each vertex $v_i \in V_r$, if there is an edge in G from v_i to $v_{\phi(i)}$, then increase the weight of v_i in G_{odd} by $w_{i\phi(i)}$.

- **Edge weight adjustment rule:** For each pair of distinct vertices v_i, $v_j \in V_r$ and $i < j$, if there is an edge $(v_i, v_{\phi(j)})$ in G, add an edge (v_i, v_j) with weight $-w_{i,\phi(j)}$ to G_{odd} (if there is already an edge (v_i, v_j) in G_{odd}, subtract $w_{i,\phi(j)}$ from its weight).

Delete from G_{odd} any edge that ends up with weight zero.

For any eigenvector $\mathbf{u}$ of B_{odd} we construct an odd eigenvector $\mathbf{x}$ for B as follows: for each vertex $v_i \in V_f$, $x_i = 0$; for each vertex $v_j \in V_r$, $x_j = u_j$ and $x_{\phi(j)} = -u_j$.

G_{even} is a weighted graph on $V_r \cup V_f$. Start with the subgraph of G induced by $V_r \cup V_f$, with the weight of vertex $v_i \in V_r \cup V_f$ equal to its degree in G and the weight of each edge (v_i, v_j) equal to its weight in G. Adjust the weights according to the following three rules:

- **Vertex weight adjustment rule:** For each vertex $v_i \in V_r$, if there is an edge in G from v_i to $v_{\phi(i)}$, then decrease the weight of v_i in G_{even} by $w_{i\phi(i)}$.

- **V_r-to-V_f Edge weight adjustment rule:** For each edge (v_i, v_j) in G_{even} such that $v_i \in V_r$ and $v_j \in V_f$, multiply its weight by $\sqrt{2}$.

- **V_r-to-V_r Edge weight adjustment rule:** For each pair of distinct vertices v_i, $v_j \in V_r$ and $i < j$, if there is an edge $(v_i, v_{\phi(j)})$ in G, add an edge (v_i, v_j) with weight $w_{i,\phi(j)}$ to G_{even} (if there is already an edge (v_i, v_j) in G_{even}, add $w_{i,\phi(j)}$ to its weight).

Delete from G_{even} any edge that ends up with weight zero.

For any eigenvector $\mathbf{u}$ of B_{even} we construct an even eigenvector $\mathbf{x}$ for B as follows: for each vertex $v_i \in V_f$, $x_i = \sqrt{2} \cdot u_i$; for each vertex $v_j \in V_r$, $x_j = x_{\phi(j)} = u_j$.

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

LEMMA 3.1. **[Zero Entries of Path Graph Eigenvectors Lemma]** *Let B be the generalized Laplacian matrix 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 $\mathbf{x} = 0$.*

Since eigenvectors are by definition not equal to the zero vector, the theorem above implies that for eigenvectors of the generalized Laplacian B of any weighted path graph, neither the first nor the last entry is zero. Likewise, such an eigenvector cannot have two consecutive zero entries.

LEMMA 3.2. *All eigenvalues of the generalized Laplacian B of a weighted path graph G on n vertices are simple (i.e., have multiplicity one).*

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

LEMMA 3.3. *The eigenvector $\mathbf{u}_2$ corresponding to the second smallest eigenvalue λ_2 of the Laplacian B of a path graph G on n vertices is odd.*

4 A Bad Family of Bounded-Degree Planar Graphs for the Simple Spectral Algorithm

In this section we present a family of bounded-degree planar graphs that have constant-size separators. However, the separators for these graphs produced by the simple spectral method have size $\Theta(n)$ for both edge and vertex separators. Since there are algorithms that produce $O(\sqrt{n})$ vertex separators for planar graphs, and hence $O(\sqrt{n})$ edge separators for bounded-degree planar graphs, the simple spectral method performs poorly on these graphs relative to other algorithms.

The family of graphs is parameterized on the positive integers. G_k consists of two path graphs each on $3k$ vertices, with a set of edges between the two paths as follows: label the vertices of one path from 1 to $3k$ in order (the **upper path**), and label the other path from $3k+1$ to $6k$ in order (the **lower path**). For $1 \leq i \leq k$ there is an edge from vertex $2k + i$ to $5k + i$. This is shown in Figure 1. It is obvious that G_k is planar for any k, and that the maximum degree of any vertex is 3.

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 will allow us to refer easily 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" will be used with respect to this automorphism. Thus, an even vector $\mathbf{x}$ has $x_i = x_{3k+i}$ for all i in the range $1 \leq i \leq 3k$; an odd vector $\mathbf{y}$ has $y_i = -y_{3k+i}$ for all i, $1 \leq i \leq 3k$.

We can now discuss the structure of the eigenvectors of the Laplacian of G_k.

LEMMA 4.1. *Any eigenvector $\mathbf{u}_i$ with eigenvalue λ_i of the Laplacian B_k of G_k can be expressed in terms of linear combinations of even and odd eigenvectors with eigenvalue λ_i. In particular, $\mathbf{u}_i$ 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 $3k$ 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 $3k$ vertices for which the vertex weights of v_{2k+1} through v_{3k} have been increased by 2.*

Proof. The first claim follows by the Even-Odd Eigenvector Theorem applied with respect to the automorphism that maps the vertices of the upper path to the vertices of the lower path and vice versa.

The second claim follows by a direct application of the construction used in the proof of the Even-Odd Graph Decomposition Theorem with respect to the same automorphism.

We now give the proof that the simple spectral

method gives bad separators for the family of graphs G_k.

THEOREM 4.1. *The simple spectral method 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 will split 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 will 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 $\mathbf{y}$ perpendicular to $\vec{1}$ ($\vec{1}$ is the smallest even eigenvector). To do this, we need to show that $\frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ is less than the second smallest even eigenvalue. From Lemma 4.1 above, we know that the second smallest even eigenvalue of B_k is the same as the second smallest eigenvalue of the Laplacian of a path graph on $3k$ vertices; it is well-known that this value is $4\sin^2(\frac{\pi}{6k})$ (see for example [Moh88]).

Let $\mathbf{z}$ be the eigenvector corresponding to the second smallest eigenvalue μ_2 for the Laplacian B of the path graph G on $4k$ vertices ($\mu_2 = 4\sin^2(\frac{\pi}{8k})$). Construct $\mathbf{x}$ as follows:

$$x_i = \begin{cases} z_i & 1 \le i \le 2k, \\ z_{7k-i+1} & 3k+1 \le i \le 5k, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

That is, we assign the first $2k$ values from the path G to the upper antenna of the roach, working in the direction towards the body, and we assign the last $2k$ 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 perpendicular to the "all-ones" vector, so is $\mathbf{x}$.

In order to see that $\mathbf{x}^T B_k \mathbf{x} < \mathbf{z}^T B \mathbf{z}$, recall that $\mathbf{y}^T A \mathbf{y} = \sum_{(v_i,v_j)\in E}(y_i - y_j)^2$ for Laplacian A. For every edge in G except one, there is an edge in G_k that contributes the same value. The one exception is the edge (v_{2k}, v_{2k+1}). Since $\mathbf{z}$ is an odd vector by Lemma 3.3, and since $\mathbf{z}$ has $4k$ entries, $z_{2k} = -z_{2k+1}$. By the Zero Entries of Path Graph Eigenvectors Lemma (Lemma 3.1), it is not possible for both z_{2k} and z_{2k+1} to be zero, so z_{2k} 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_{2k}, v_{2k+1}) and (v_{5k}, v_{5k+1}). Each of these edges contributes c^2 to $\mathbf{x}^T B_k \mathbf{x}$. Thus we have

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

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

$$\lambda_2(G_k) \le \frac{\mathbf{x}^T B_k \mathbf{x}}{\mathbf{x}^T \mathbf{x}} < \frac{\mathbf{z}^T B \mathbf{z}}{\mathbf{z}^T \mathbf{z}} = 4\sin^2(\frac{\pi}{8k}) < 4\sin^2(\frac{\pi}{6k}).$$

That is, the second smallest eigenvalue of B_k is less than any non-zero even eigenvalue, and is thus odd by the Even-Odd Eigenvector Theorem (Theorem 3.2).

There are a few details to finish off. In particular, we need to show that there aren't too many zero entries in $\mathbf{u}_2$ (the simple spectral method as defined in this paper won't separate vertices with the same value). Since $\mathbf{u}_2$ has no even component, Lemmas 3.1 (the Zero Entries of Path Graph Eigenvectors Lemma) and 4.1 imply that $\mathbf{u}_2$ cannot have consecutive zeros, and the values corresponding to vertices $3k$ and $6k$ are non-zero. Thus the edge separator generated by the simple spectral method 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. The theorem follows.

5 A Bad Family of Graphs for the Modified Simple Spectral Algorithm

While the roach graph defeats a simple spectral bisection method, the second smallest eigenvector can still be used to find a small separator using the modified simple spectral algorithm. In particular, Theorem 3.1 implies that considering all threshold cuts induced by $\mathbf{u}_2$ will let us find a constant-size cut: If q_{min} is the minimum cut quotient for these cuts, then

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

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 the next section we define the Tree-Cross-Path Graph, for which the modified simple spectral algorithm does poorly.

5.1 The Tree-Cross-Path Graph

Consider a graph that consists of two complete binary trees of k levels for some $k > 0$, connected by an edge between their respective roots. We will refer to such a graph as a **double tree**.

We call a graph consisting of the crossproduct of a double tree on p_1 vertices and a path graph on p_2 vertices a tree-cross-path graph. We will show below that there is a tree-cross-path graph that will defeat the modified simple spectral algorithm. In order to show

that, we will need to understand the structure of the eigenvectors of such a graph; that requires that we have bounds on the size of λ_2 for trees and double trees.

5.2 Bounds on λ_2 for Trees and Double Trees To bound the size of λ_2 of a tree, we first apply the Even-Odd Decomposition Theorem (Theorem 3.3) a number of times to show that the eigenvalues of a balanced binary tree can be computed from a few simple types of weighted graphs. We then bound the eigenvalues for these types of graphs using the interleaving theorems and matrix perturbation techniques. We show that for a complete binary tree on n vertices, $\lambda_2 = \Theta(\frac{1}{n})$. More specifically, $\frac{1}{n} < \lambda_2 < \frac{2}{n}$. We state this result in the form of a lemma; the proof is in the complete version of the paper.

LEMMA 5.1. *For a complete balanced binary tree on n vertices, we have $\frac{1}{n} < \lambda_2 < \frac{2}{n}$.*

For double trees where each of the component trees has k levels and $n = 2^{k+1} - 2$ vertices, we have the following lemma:

LEMMA 5.2. *For a double tree on n vertices, we have $\frac{1}{n} \leq \lambda_2 < \frac{4}{n}$.*

The proof of this lemma is also given in the complete version of the paper.

We can now formally state the result for this section:

THEOREM 5.1. *There exists a graph G for which the modified simple spectral 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 the theoretical upper bound provided by 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 we have integer sizes for the tree and the path, we restrict p to integers of the form $2^k - 2$ for $k > 2$. Then we choose c in the range specified such that $cp^{\frac{1}{2}}$ is an integer (by our choice of p there will be 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 will be μ_2 (i.e., μ_2 added to the zero eigenvalue of the double tree). But we have that $\mu_2 = 4\sin^2(\frac{\pi}{2cp^{\frac{1}{2}}})$, and that ν_2 is greater than or equal to $\frac{1}{p}$. Therefore we need 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 $0 < \theta \leq \frac{\pi}{2}$, we want

$$\frac{\pi}{2cp^{\frac{1}{2}}} < \frac{1}{2p^{\frac{1}{2}}}, \text{ or } \pi < c.$$

Clearly by our choice of c this inequality holds.

Since path graph eigenvalues are simple (Lemma 3.2), the second smallest eigenvalue of G will also be simple.

We 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 us 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 the eigenvalue is simple.

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 modified simple spectral method 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, we have that

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

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

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

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

6 A Bad Family of Graphs for Generalized Spectral Algorithms

The results of the previous section can be extended to more general 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 separator ratio for all possible breaks and choosing the best one is allowed.

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

The following theorem gives a bound on how well such algorithms do when the number of eigenvectors used is a constant:

THEOREM 6.1. *Consider the purely spectral algorithms that uses 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 will produce a separator S for G such that $|S| \geq \left(\frac{|S^*|}{\pi k+1}\right)^2$.*

Proof. We will 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 $2^k - 2$) 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.

Recall that the eigenvalues of a graph crossproduct are all pairwise sums of the eigenvalues from the graphs used in the crossproduct operation. Assume for the moment that the k smallest nonzero eigenvalues of any $G \in \mathcal{G}$ are the same as the k smallest nonzero eigenvalues of the path graph used in defining G. This will clearly hold if we can show that these k eigenvalues are smaller than λ_2 of the double tree; in that case the k smallest non-zero eigenvalues of the crossproduct will be these eigenvalues from the path graph added to the zero eigenvalue of the double tree. Since the path graph eigenvalues are simple (Lemma 3.2), the corresponding tree-cross-path eigenvalues will also be simple.

We 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 us to construct an eigenvector for each of these k tree-cross-path eigenvalues as follows: Assign each vertex in double tree copy i the value for vertex i in the path graph eigenvector for this eigenvalue. Note that these are the only possible eigenvectors since these eigenvalues are simple.

The purely spectral algorithm will produce a cut S with cut quotient q. Recall our assumption about the k smallest eigenvectors and consider any copy of the double tree: since every vertex in that copy gets the same value for each eigenvector, the same value will be assigned to each vertex in this copy by the algorithm. This implies that S must separate at least two copies of the double tree, and thus must cut at least p edges.

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

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

since we have $c \leq \pi k + 1$, the claim in the theorem statement holds if our assumption holds.

To prove that the assumption about the form of the k smallest eigenvectors holds for all $G \in \mathcal{G}$, we still need to prove that a path graph on $cp^{\frac{1}{2}}$ vertices has k nonzero eigenvalues smaller than λ_2 for a double tree on p vertices. Recall that the eigenvalues of a path graph on l vertices are $4\sin^2(\frac{\pi i}{2l})$ for $0 \leq i < l$, and that λ_2 for a double tree on p vertices is greater than or equal to $\frac{1}{p}$. Therefore we need to show that

$$4\sin^2\left(\frac{\pi k}{2cp^{\frac{1}{2}}}\right) < \frac{1}{p}.$$

Reorganizing, simplifying, and noting that $\sin(\theta) < \theta$ for $0 < \theta \leq \frac{\pi}{2}$, we have

$$\frac{\pi k}{2cp^{\frac{1}{2}}} < \frac{1}{2p^{\frac{1}{2}}}, \text{ or } \pi k < c.$$

Clearly this inequality holds.

Note that for the case in which k is constant, we have the following results:

- the cut quotient q_S will be no better than the best cut quotient q_{min} produced by looking at 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.

These results can be shown using techniques from the previous section. Thus, G is a graph for which using k eigenvectors does not improve the performance of the modified simple spectral algorithm.

6.1 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 are allowed to 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 6.2. *For large enough 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 will produce a separator S for G that has a cut quotient greater than $i(G)$ by at least a factor of $n^{\left(\frac{1}{4}-\epsilon\right)} - 1$.*

The proof of this theorem is in the complete version of this paper.

6.2 A Final Note About Tree-Cross-Path Graphs While the tree-cross-path graph appears to be very specialized, we can make the following argument that it has some relation to practice: We noted in Section 3 that the Second Interleaving Theorem implies that adding an edge to a graph G gives a new graph G' with λ_2' greater than or equal to λ_2 for G. Therefore the preceding results holds if we replace each tree in the double trees with a graph that has a complete binary tree as a spanning tree (the edge between the two graphs will be between the vertices at the roots of the spanning trees, and connections between copies of the "double graphs" in the crossproduct will be between corresponding vertices in the spanning trees). We could therefore construct a three-dimensional finite element mesh from our example that would represent the channel tunnel (the Chunnel) between England and France; the chunnel is a pair of long tubes with a small connection between the center.

References

[AG84] B. Aspvall and J. R. Gilbert. Graph coloring using eigenvalue decomposition. *SIAM Journal on Algebraic and Discrete Methods*, 5(4):526–538, December 1984.

[AGM87] N. Alon, Z. Galil, and V. D. Milman. Better expanders and superconcentrators. *Journal of Algorithms*, 8:337–347, 1987.

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38:73–88, 1985.

[Bar82] Earl R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM J. Alg. Disc. Meth.*, 3(4):541–550, December 1982.

[Bop87] R. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *28th Annual Symposium on Foundations of Computer Science*, pages 280–285, Los Angeles, October 1987. IEEE.

[CDS79] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs*. Academic Press, New York, 1979.

[DH73] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Develop.*, 17:420–425, 1973.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[Fie75] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(100):619–633, 1975.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–completeness*. Freeman, San Francisco, 1979.

[GL89] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1989.

[HK92] Lars Hagen and Andrew B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEE Transactions on Computer-Aided Design*, 11(9):1074–1085, September 1992.

[LR88] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on the Foundations of Computer Science*, pages 422–431. IEEE Computer Society, October 1988.

[Moh88] B. Mohar. The laplacian spectrum of graphs. In *Sixth International Conference on the Theory and Applications of Graphs*, pages 871–898, 1988.

[Moh89] Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47:274–291, 1989.

[PSL90] Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, July 1990.

[Sim91] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2/3):135–148, 1991.

[Wil65] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, 1965.