

## ISOPERIMETRIC PARTITIONING: A NEW ALGORITHM FOR GRAPH PARTITIONING\*

LEO GRADY<sup>†</sup> AND ERIC L. SCHWARTZ<sup>‡</sup>

**Abstract.** We present a new algorithm for graph partitioning based on optimization of the combinatorial isoperimetric constant. It is shown empirically that this algorithm is competitive with other global partitioning algorithms in terms of partition quality. The isoperimetric algorithm is easy to parallelize, does not require coordinate information, and handles nonplanar graphs, weighted graphs, and families of graphs which are known to cause problems for other methods. Compared to spectral partitioning, the isoperimetric algorithm is faster and more stable. An exact circuit analogy to the algorithm is also developed with a natural random walks interpretation. The isoperimetric algorithm for graph partitioning is implemented in our publicly available Graph Analysis Toolbox [L. Grady, Ph.D. thesis, Boston University, Boston, MA, 2004], [L. Grady and E. L. Schwartz, Technical report TR-03-021, Boston University, Boston, MA, 2003] for MATLAB obtainable at <http://eslab.bu.edu/software/graphanalysis/>. This toolbox was used to generate all of the results compiled in the tables of this paper.

**Key words.** graph partitioning, spectral partitioning, isoperimetric problem, graph theory, linear equations

**AMS subject classifications.** 91B28, 91B30, 60J75, 30A36

**DOI.** 10.1137/040609008

**1. Introduction.** In the graph partitioning problem, one chooses subsets of the vertex set of a graph such that the sets share a minimal number of spanning edges while satisfying a specified cardinality constraint. Applications of graph partitioning include parallel processing [63], solving sparse linear systems [58], VLSI circuit design [4], and image segmentation [61, 68, 32].

Methods of graph partitioning take different forms, depending on the number of partitions required—whether or not the nodes have coordinates—and the cardinality constraints of the sets. In this paper, we use the term *partition* to refer to the assignment of each node in the vertex set into two (not necessarily equal) parts. We propose an algorithm termed *isoperimetric partitioning*, since it is derived and motivated by the isoperimetric constant defined for continuous manifolds [15]. The isoperimetric algorithm most closely resembles spectral partitioning in its use and ability to create hybrids with other algorithms (e.g., multilevel spectral partitioning [40] and geometric-spectral partitioning [14]). However, it requires the solution to a large, sparse system of equations rather than solving the eigenvector problem for a large, sparse matrix. This difference leads to improved speed and numerical stability.

The paper is organized as follows: we begin by deriving the isoperimetric algorithm from the isoperimetric constant of a graph in section 2, followed in section 3 by two physical analogies for the isoperimetric algorithm, and in section 4 by proving a few formal properties of the algorithm. In section 5, we review the most popular and

---

\*Received by the editors May 25, 2004; accepted for publication (in revised form) February 25, 2005; published electronically February 3, 2006. This work was supported in part by the Office of Naval Research (ONR N00011401-1-0624).

<http://www.siam.org/journals/sisc/27-6/60900.html>

<sup>†</sup>Department of Imaging and Visualization, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540 (leo.grady@siemens.com).

<sup>‡</sup>Departments of Cognitive and Neural Systems and Electrical and Computer Engineering, Boston University, Boston, MA 02215 (eric@bu.edu).

effective graph partitioning algorithms and the relation of the present work. Section 6 provides examples intended to build intuition about the algorithm behavior. Section 7 validates the present method on various general types of graphs and several specific graphs, followed by the conclusion.

**2. Isoperimetric algorithm.** A *graph* is a pair  $G = (V, E)$  with vertices  $v \in V$  and edges  $e \in E \subseteq V \times V$ . An edge,  $e$ , spanning two vertices,  $v_i$  and  $v_j$ , is denoted by  $e_{ij}$ . Let  $n = |V|$  and  $m = |E|$ , where  $|\cdot|$  denotes cardinality. A *weighted graph* has a value (here assumed to be nonnegative and real) assigned to each edge called a *weight*. The weight of edge  $e_{ij}$  is denoted by  $w(e_{ij})$  or  $w_{ij}$ . Since weighted graphs are more general than unweighted graphs (i.e.,  $w(e_{ij}) = 1$  for all  $e_{ij} \in E$  in the unweighted case), we develop all our results for weighted graphs.

Graph partitioning has been strongly influenced by properties of a combinatorial formulation of the classic isoperimetric problem: *For a fixed area, find the shape with minimum perimeter* [16]. The approach to graph partitioning presented here is a polynomial time heuristic for the NP-hard [55] problem of finding a graph with minimum perimeter for a fixed area.

Cheeger defined [15] the *isoperimetric constant*  $h$  of a manifold as

$$(2.1) \quad h = \inf_S \frac{|\partial S|}{\text{Vol}_S},$$

where  $S$  is a region in the manifold,  $\text{Vol}_S$  denotes the volume of region  $S$ ,  $|\partial S|$  is the area of the boundary of region  $S$ , and  $h$  is the infimum of the ratio over all possible  $S$ . For a compact manifold,  $\text{Vol}_S \leq \frac{1}{2} \text{Vol}_{\text{Total}}$ , and for a noncompact manifold,  $\text{Vol}_S < \infty$  (see [55, 54]).

For a graph,  $G$ , the *isoperimetric number* [55],  $h_G$ , is

$$(2.2) \quad h_G = \inf_S \frac{|\partial S|}{\text{Vol}_S},$$

where  $S \subset V$  and

$$(2.3) \quad \text{Vol}_S \leq \frac{1}{2} \text{Vol}_V.$$

In finite graphs, the infimum in (2.2) becomes a minimum. The boundary of a set,  $S$ , is defined as  $\partial S = \{e_{ij} | i \in S, j \in \bar{S}\}$  and on a weighted graph

$$(2.4) \quad |\partial S| = \sum_{e_{ij} \in \partial S} w(e_{ij}).$$

In the context of graph partitioning, combinatorial volume is typically taken as

$$(2.5) \quad \text{Vol}_S = |S|.$$

For a given set of nodes,  $S$ , we term the ratio of its boundary to its volume as the *isoperimetric ratio* and denote it by  $h(S)$ . The *isoperimetric sets* for a graph,  $G$ , are any  $S$  and  $\bar{S}$  for which  $h(S) = h_G$ . The specification of a set satisfying (2.3), together with its complement, may be considered as a *partition*, and therefore we use the term interchangeably with the specification of a set satisfying (2.3). A good partition is defined to be one with a low isoperimetric ratio (i.e., the optimal partition consists of the isoperimetric sets themselves). Therefore, our goal is to maximize  $\text{Vol}_S$

while minimizing  $|\partial S|$ . Finding isoperimetric sets is an NP-hard problem [55], so our algorithm is a heuristic for finding a set with a low isoperimetric ratio that runs in polynomial time.

Define an indicator vector that takes a binary value at each node

$$(2.6) \quad x_i = \begin{cases} 0 & \text{if } v_i \in S, \\ 1 & \text{if } v_i \in \bar{S}. \end{cases}$$

A specification of  $x$  also defines a partition. Define the  $n \times n$  matrix  $L$  of a graph as

$$(2.7) \quad L_{v_i v_j} = \begin{cases} d_i & \text{if } i = j, \\ -w(e_{ij}) & \text{if } e_{ij} \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where  $d_i$  denotes the weighted *degree* of vertex  $v_i$

$$(2.8) \quad d_i = \sum_{e_{ij}} w(e_{ij}) \quad \forall e_{ij} \in E.$$

The notation  $L_{v_i v_j}$  is used to indicate that the matrix  $L$  is being indexed by vertices  $v_i$  and  $v_j$ . This matrix is also known as the *admittance matrix* in the context of circuit theory or the *Laplacian matrix* (see [51] for a review).

By definition of  $L$

$$(2.9) \quad |\partial S| = x^T L x,$$

and  $\text{Vol}_S = x^T r$ , where  $r$  denotes the vector of all ones. Maximizing the volume of  $S$  subject to  $\text{Vol}_S = k$  for some constant  $k \leq \frac{1}{2} \text{Vol}_V$  may be done by asserting the constraint

$$(2.10) \quad x^T r = k.$$

In terms of the indicator vector, the isoperimetric number of a graph (2.2) is given by

$$(2.11) \quad h_G = \min_x \frac{x^T L x}{x^T r}.$$

Given an indicator vector,  $x$ , then  $h(x)$  is used to represent the isoperimetric ratio associated with that partition.

The constrained optimization of the isoperimetric ratio is made into a free variation via the introduction of a Lagrange multiplier [6],  $\Lambda$ , and relaxation of the binary definition of  $x$  to take nonnegative real values. Therefore, solving for an optimal partition may be accomplished by minimizing the function

$$(2.12) \quad Q(x) = x^T L x - \Lambda(x^T r - k).$$

Since  $L$  is positive semidefinite (see [8, 26]) and  $x^T r$  is nonnegative,  $Q(x)$  will be at a minimum at its critical points. Differentiating  $Q(x)$  with respect to  $x$  yields

$$(2.13) \quad \frac{dQ(x)}{dx} = 2Lx - \Lambda r.$$

Thus, the problem of finding the  $x$  that minimizes (2.12) (i.e., the minimal partition) reduces to solving the linear system

$$(2.14) \quad 2Lx = \Lambda r.$$

Henceforth, the scalar multiplier 2 and the scalar  $\Lambda$  are ignored since, as will be seen later, we are only concerned with the relative values of the solution.

The matrix  $L$  is singular: all rows and columns sum to zero (i.e., the vector  $r$  spans its nullspace), so finding a unique solution to (2.14) requires an additional constraint.

The graph is assumed to be connected, since the optimal partitions are clearly each connected component (i.e.,  $h(x) = h_G = 0$ ) if the graph is disconnected. A linear time breadth-first search may be performed to check for connectivity of the graph. Note that, in general, a graph with  $c$  connected components will correspond to a matrix  $L$  with rank  $(n - c)$  [8]. If a node,  $v_g$ , is arbitrarily designated to include in  $S$  (i.e., fix  $x_g = 0$ ), this is reflected in (2.14) by removing the  $g$ th row and column of  $L$ , and the  $g$ th row of  $x$  and  $r$  such that

$$(2.15) \quad L_0 x_0 = r_0,$$

where  $L_0$  indicates the Laplacian with a row/column removed,  $x_0$  is the vector  $x$  with the corresponding removed entry, and  $r_0$  is  $r$  with the removed row. Note that (2.15) is a nonsingular system of equations.

Solving (2.15) for  $x_0$  yields a nonnegative, real-valued solution that may be converted into a partition by setting a threshold. Nodes with an  $x_i$  below the threshold are placed in  $S$  and nodes with an  $x_j$  above the threshold are placed in  $\bar{S}$ . Setting a threshold is referred to as a *cut* since it divides the nodes into  $S$  and  $\bar{S}$ . We use  $x$  to collectively refer to the  $x_0$  and the designated  $x_g = 0$  value. As in spectral partitioning, common methods [64] of defining a threshold are the *median cut*, which chooses the median value of  $x$  as the threshold (thereby guaranteeing  $|S| = |\bar{S}|$ ), the *jump cut*, which chooses a threshold that separates nodes on either side of the largest “jump” in a sorted  $x$ , and the *criterion cut*, which chooses the threshold that gives the lowest value of  $h(x)$  (called the “ratio cut” in [64]).

**2.1. Algorithmic details.** The isoperimetric algorithm for partitioning a graph may be summarized as follows:

1. Choose a ground node,  $v_g$ .
2. Solve (2.15).
3. Cut  $x$  based on the method of choice to obtain  $S$  and  $\bar{S}$ .

There are several possible strategies for choosing the ground node. Anderson and Morley [5] proved that the spectral radius of  $L$ ,  $\rho(L)$ , satisfies  $\rho(L) \leq 2d_{\max}$ , suggesting that grounding the node of maximum degree may have the most beneficial effect on the conditioning of (2.15). In the comparison section of this paper, we employ two grounding strategies (grounding the maximum degree node and grounding a random node).

The main computational hurdle of the isoperimetric algorithm is the solution of (2.15). An advantage of using the method of conjugate gradients is that an efficient parallelization of this technique is known [20, 33], suggesting that the majority of computation in the isoperimetric algorithm may be parallelized. In fact, cheap parallel machines in the form of commodity graphics hardware have already proven to be effective in executing the conjugate gradients method [9].

**3. Physical analogies.** The solution to (2.15) may be interpreted as the solution to a set of electrical potentials of a particular circuit and, with a slight change, in the context of a random walk. These analogies are primarily introduced in order to guide intuition about the behavior of the algorithm and will be relied upon in section 6 to provide understanding of the examples contained there. However, an added benefit of the circuit analogy is the adoption of terminology used to describe the node fixed to zero and the solution to (2.15).

**3.1. Circuit analogy.** Equation (2.14) occurs in circuit theory when solving for the electrical potentials of an ungrounded circuit in the presence of current sources [11, 65]. After grounding a node in the circuit (i.e., fixing its potential to zero), determination of the remaining potentials requires a solution of (2.15). Therefore, we refer to the node,  $v_g$ , for which we set  $x_g = 0$  as the *ground node*. Likewise, the solution,  $x_i$ , obtained from (2.15) at node  $v_i$ , will be referred to as the *potential* for node  $v_i$ . The need for fixing an  $x_g = 0$  to constrain (2.14) may be seen not only from the necessity of grounding a circuit powered only by current sources in order to find unique potentials, but also from the need to provide a boundary condition in order to find a solution to Poisson's equation, of which (2.14) is a combinatorial analogue. In our case, the "boundary condition" is that the grounded node is fixed to zero.

Define the  $m \times n$  edge-node *incidence matrix* as

$$(3.1) \quad A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise} \end{cases}$$

for every vertex  $v_k$  and edge  $e_{ij}$ , where  $e_{ij}$  has been arbitrarily assigned an orientation. As with the Laplacian matrix,  $A_{e_{ij}v_k}$  is used to indicate that the incidence matrix is indexed by edge  $e_{ij}$  and node  $v_k$ . As an operator,  $A$  may be interpreted as a combinatorial gradient operator and  $A^T$  as a combinatorial divergence [10, 65]. The  $m \times m$  *constitutive matrix*,  $C$ , is the diagonal matrix with the weights of each edge along the diagonal.

As in the familiar continuous setting, the combinatorial Laplacian is equal to the composition of the combinatorial divergence operator with the combinatorial gradient operator,  $L = A^T A$ . The constitutive matrix defines a weighted inner product of edge values, i.e.,  $\langle y, Cy \rangle$  for a vector of edge values,  $y$  [65, 11]. Therefore, the combinatorial Laplacian operator generalizes to the combinatorial Laplace–Beltrami operator via  $L = A^T C A$ . The case of a uniform (unit) metric (i.e., equally weighted edges) reduces to  $C = I$  and  $L = A^T A$ . Removing a column of the incidence matrix produces what is known as the *reduced incidence matrix*,  $A_0$  [27].

With this interpretation of the notation used above, the three fundamental equations of circuit theory (Kirchhoff's current and voltage laws and Ohm's law) may be written for a grounded circuit as

$$(3.2) \quad A_0^T y = f \quad (\text{Kirchhoff's current law}),$$

$$(3.3) \quad Cp = y \quad (\text{Ohm's law}),$$

$$(3.4) \quad p = A_0 x \quad (\text{Kirchhoff's voltage law})$$

for a vector of branch currents,  $y$ , current sources,  $f$ , and potential drops (voltages),  $p$  [11]. Note that there are no voltage sources present in this formulation. These three equations may be combined into the linear system

$$(3.5) \quad A_0^T C A_0 x = L_0 x = f,$$

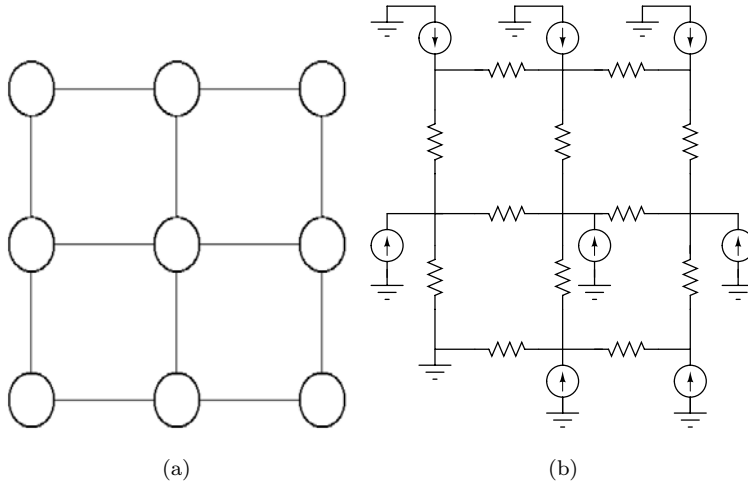


FIG. 3.1. An example of a simple graph (a), and its equivalent circuit (b). Solving (2.15) (using the node in the lower left as ground) for the graph depicted in (a) is equivalent to connecting (b) and measuring the potential values at each node.

since  $A^T C A = L$  [8].

In summary, the solution to (2.15) in the isoperimetric algorithm is provided by the steady state of a circuit where each edge has a conductance equal to the edge weight and each node is attached to a current source of magnitude equal to the degree (i.e., the sum of the conductances of incident edges) of the node. The potentials that are established on the nodes of this circuit are exactly those which are being solved for in (2.15). An example of this equivalent circuit is displayed in Figure 3.1.

One final remark on the circuit analogy to (2.15) follows from Maxwell's principle of least dissipation of power: a circuit with minimal power dissipation provides a solution to Kirchhoff's current and voltage laws [50]. Explicitly, solving (2.15) for  $x$  is equivalent to solving the dual equation for  $y = C A x$ . The power of the equivalent circuit is  $P = I^2 R = y^T C^{-1} y$  subject to the constraint from Kirchhoff's law that  $A^T y = f$ . Therefore, the  $y$  found via  $y = C A x$  also minimizes the above expression [65, 7]. Thus, our approach to minimizing the combinatorial isoperimetric ratio is identical to minimizing the power of the equivalent electrical circuit with the specified current sources and ground [65].

**3.2. Random walk interpretation.** In section 2, the combinatorial "volume" of a node set was defined by (2.5) as the cardinality of the set. However, this definition of combinatorial volume can lead to strange conclusions when trying to develop the general theory of Riemannian manifolds on a combinatorial space [17, 18, 54]. The alternate definition proposed by [18] defines volume as

$$(3.6) \quad \text{Vol}_S = \sum_{v_i \in S} d_i.$$

Using this notion of volume in the formulation of the isoperimetric constant in (2.2) yields the problem studied under the name of the *minimum quotient cut* [48].

Substituting this notion of volume (which is equivalent up to a scale factor on a

regular graph) into the above development leads to the solution of

$$(3.7) \quad L_0 x_0 = d_0,$$

where  $d_0$  is the reduced vector of node degrees.

The solution,  $x_i$ , of (3.7) may be interpreted [67, 21] as the expected number of steps taken by a random walker leaving node  $v_i$  before it reaches the ground node, where the transition probability between nodes  $v_i$  and  $v_j$  is derived from the weights as  $p_{ij} = \frac{w_{ij}}{d_i}$ .

We note that, in the formulation of their Laplacian matrix, this same notion of “volume” was effectively used in the domain of image segmentation by Shi and Malik’s “normalized cuts” algorithm [61].

**4. Some formal properties of the algorithm.** In this section, we prove one property of the algorithm and examine the behavior of the solution on two classes of graphs: trees and fully connected graphs.

**4.1. Connectivity.** It is known that at least one of the isoperimetric sets (if not unique) is such that both partitions are connected [55]. Therefore, it is of interest to examine the connectivity properties of a partitioning algorithm. Fiedler proved that partitioning a graph by thresholding the values in the Fiedler vector is guaranteed to produce connected partitions [25]. In this section, we examine the connectivity properties of the partitions obtained by thresholding the potentials solved for in (2.15) (or (3.7)). We prove that the partition containing the grounded node (i.e., the set  $S$ ) must be connected, regardless of how a threshold (i.e., cut) is chosen. The strategy for establishing this will be to show that every node has a path to ground with a monotonically decreasing potential. We note that the partition not containing the ground may or may not be connected (cf. Figures 6.1 and 6.2).

**PROPOSITION 4.1.** *If the set of vertices,  $V$ , is connected then, for any  $\alpha$ , the subgraph with vertex set  $N \subseteq V$  defined by  $N = \{v_i \in V | x_i < \alpha\}$  is connected when  $x_0$  satisfies  $L_0 x_0 = f_0$  for any  $f_0 \geq 0$ .*

This proposition follows directly from the proof of the following lemma.

**LEMMA 4.2.** *For every node,  $v_i$ , there exists a path to the ground node,  $v_g$ , defined by  $P_i = \{v_i, v^1, v^2, \dots, v_g\}$  such that  $x_i \geq x^1 \geq x^2 \geq \dots \geq 0$ , when  $L_0 x_0 = f_0$  for any  $f_0 \geq 0$ .*

*Proof.* By (2.15) each nongrounded node assumes a potential

$$(4.1) \quad x_i = \frac{1}{d_i} \sum_{e_{ij} \in E} w_{ij} x_j + \frac{f_i}{d_i},$$

i.e., the potential of each nongrounded node is equal to a nonnegative constant added to the (weighted) average potential of its neighbors. Note that (4.1) is a combinatorial formulation of the mean value theorem [1] in the presence of sources. For any connected subset,  $S \subseteq V$ ,  $v_g \notin S$ , denote the set of nodes on the boundary of  $S$  as  $S_b \subset V$ , such that  $S_b = \{v_i | e_{ij} \in E, \exists v_j \in S, v_i \notin S\}$ .

Now, either

1.  $v_g \in S_b$ , or
2.  $\exists v_i \in S_b$ , such that  $x_i \leq \min x_j$  for all  $v_j \in S$  by (4.1), since the graph is connected.

By induction, every node has a path to ground with a monotonically decreasing potential (i.e., start with  $S = \{v_i\}$ ; add nodes with a nonincreasing potential until ground is reached).  $\square$

**4.2. Spanning trees.** Since a tree will have a unique path from any node to ground, Lemma 4.2 guarantees that the nodes in this path will have a nonincreasing potential. However, since a tree is a special case of a graph, and its reduced incidence matrix is square, there is an alternate derivation of this result. A theorem by Branin [10] shows that for node  $v_k$ , edge  $e_{ij}$ , and ground  $v_g$ , the inverse of the reduced incidence matrix for a spanning tree,  $B = A_0^{-1}$ , is

$$(4.2) \quad B_{v_k e_{ij}} = \begin{cases} +1 & \text{if } e_{ij} \text{ is positively traversed in the path from } v_k \text{ to } v_g, \\ -1 & \text{if } e_{ij} \text{ is negatively traversed in the path from } v_k \text{ to } v_g, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$(4.3) \quad L_0^{-1} = (A_0^T C A_0)^{-1} = B^T C^{-1} B.$$

Each value of  $L_0^{-1}$  may therefore be interpreted as the sum of the reciprocal weights (i.e., the *resistances*) of shared edges along the unique path to  $v_g$  between nodes  $v_i$  and  $v_j$ , i.e., the shared *distance* of the unique paths from  $v_i$  and  $v_j$  to  $v_g$  in the metric interpretation.

It follows that the potential values taken by  $x_0$  in  $x_0 = L_0^{-1} f_0$  are monotonically increasing along the path from  $v_g$  to any other node for nonnegative  $f_0$  and  $C$ .

**4.3. Fully connected graphs.** The isoperimetric algorithm will produce a solution to (2.15) that prefers each node equally when applied to fully connected graphs with uniform weights. Any set with cardinality equal to half the cardinality of the vertex set is a solution to the isoperimetric problem for a fully connected graph with uniform weights. For a uniform edge weight,  $w(e_{ij}) = \kappa$  for all  $e_{ij} \in E$ , the solution,  $x_0$ , to (2.15) will be  $x_i = 1/\kappa$  for all  $v_i \in V$ . The use of the median or criterion cut method will choose half of the nodes arbitrarily. Although it should be pointed out that using a median or criterion cut to partition a vector of randomly assigned potentials will also produce equal-sized (in the case optimal) partitions, the solution to (2.15) is unique for a specified ground (in contrast to spectral partitioning, which has  $n - 1$  solutions) and explicitly gives no preference to a node by returning all potentials as equal.

**5. Review of previous work.** Many approaches have been proposed for the graph partitioning problem and its related guises (e.g., circuit placement) in the past 50 years. Consistently, however, the most popular algorithms for graph partitioning are spectral partitioning [35] and the (multilevel) Kernighan–Lin partitioning method [41, 22].

**5.1. Algebraic methods.** Converting the partitioning problem into a system of linear equations is not new. In fact, Kodres' work [45, 44] specifies two subsets of nodes as being in different partitions. This specification converts the “minimal cut” minimization of (2.9) into a tractable problem with nontrivial (i.e., not the all-zeros) solution. The solution to the system of equations obtained by Kodres is also a real-valued solution that must be converted into a “hard” partitioning with a threshold. Similarly, the GORDIAN algorithm [42] imposes constraints on multiple nodes to obtain a linear system and is later modified [62] to include a modified cost function (weighted by a linear factor). These approaches were combined in the PARABOLI method [59], with the addition that the fixed nodes are those that obtain extremal



values from the solution to the eigenvector problem. However, as indicated by Hall [35], it is often unclear how to designate the nodes that will belong to different partitions. In contrast, our approach assigns only a single node to a partition, which would eventually be assigned to a partition anyway. In general, the multiple-node constraint (MNC) approach differs from the present approach in the following ways:

1. An MNC approach designates some set of nodes to be in opposite partitions, a priori, instead of allowing that decision to be dictated by the graph structure.
2. The isoperimetric algorithm permits one of the partitions (that which does not contain the ground) to be disconnected, while an MNC approach forces both partitions to be connected (for two constrained nodes).
3. On a regular lattice, the isoperimetric algorithm naturally imitates the known solution to the isoperimetric problem on the plane (see Figure 6.1). However, an MNC approach requires a contrived constraint placement to achieve the same.

The *skewed graph partitioning* approach pursued in [38] permits specification of affinities for each node to belong to a particular partition. Viewed in the context of skewed graph partitioning, one might interpret the isoperimetric algorithm as asking the question: given definite knowledge of the partition membership of a single node (i.e., the ground is in set  $S$ ), compute a good partitioning in which the remaining nodes are biased to be in the opposite partition (i.e., in  $\bar{S}$ ). Under this interpretation, each remaining node is given either a unity bias to belong to  $\bar{S}$  or a bias equal to the node degree, depending on the notion of volume employed. Viewed in this context, the approach taken by the isoperimetric algorithm appears reasonable—since the grounded node must fall into one of the two partitions, the problem is to determine which of the remaining nodes fall into the opposite partition.

Finally, we note that grounding node  $v_i$  and performing the matrix inversion of (2.15) may also be interpreted algebraically as computing the  $i$ th term of the generalized inverse of the Laplacian, termed the *Eichinger matrix* by Kunz [46].

**5.2. Spectral partitioning.** Building on the early work of Fiedler [24, 25, 23], Alon [3, 2], and Cheeger [15] who demonstrated the relationship between the second smallest eigenvalue of the Laplacian matrix (the *Fiedler value*) for a graph and its isoperimetric number, spectral partitioning was one of the first graph partitioning algorithms to be successful [35, 19, 57]. The algorithm partitions a graph by finding the eigenvector corresponding to the Fiedler value, termed the *Fiedler vector*, and cutting the graph based on the value in the Fiedler vector associated with each node. A physical analogy for the Fiedler vector is the second harmonic of a vibrating surface. Like isoperimetric partitioning, the output of spectral partitioning is a set of values assigned to each node, which allows a cut to be a perfect bisection by choosing a zero threshold (the median cut) or by choosing the threshold that generates a partition with the best isoperimetric ratio (the criterion cut). The flexibility of spectral partitioning allows it to be used as a part of hybridized graph partitioning algorithms, such as geometric-spectral partitioning [14] and multilevel approaches [37, 12].

Spectral partitioning attempts to minimize the isoperimetric ratio of a partition by solving

$$(5.1) \quad Lz = \lambda z,$$

with  $L$  defined as above and  $\lambda$  representing the Fiedler value. Since the vector of all ones,  $r$ , is an eigenvector corresponding to the smallest eigenvalue (zero) of  $L$ , the goal is to find the eigenvector associated with the second smallest eigenvalue of  $L$ .

Requiring  $z^T r = 0$  and  $z^T z = n$  may be viewed as additional constraints employed in the derivation of spectral partitioning to circumvent the singularity of  $L$  (for an explicit formulation of spectral partitioning from this viewpoint, see [39]). Therefore, one way of viewing the difference between the isoperimetric and the spectral methods is in the choice of the additional constraint that regularizes the singular nature of the Laplacian  $L$ .

In the context of spectral partitioning, the indicator vector  $z$  is usually defined as

$$(5.2) \quad z_i = \begin{cases} -1 & \text{if } v_i \in \bar{S}, \\ +1 & \text{if } v_i \in S, \end{cases}$$

such that  $z$  is orthogonal to  $r$  for an equal-sized partition. The two definitions of the indicator vector ((2.6) and (5.2)) are related through  $x = \frac{1}{2}(z + r)$ . Since  $r$  is in the nullspace of  $L$ , these definitions are equivalent up to a scaling.

Simple, unhybridized, unilevel spectral partitioning lags behind modern multi-level algorithms (e.g., [37]) in terms of partition quality. However, compared with other global graph partitioning algorithms, spectral partitioning still performs well. Furthermore, spectral partitioning is one of the only approaches that does not require prior specification of node coordinates or the partition cardinalities, instead offering the flexibility to automatically choose the partition cardinalities that offer the best cut.

The major drawbacks of spectral partitioning are its speed and numerical stability. Even using the Lanczos algorithm [30] to find the Fiedler vector for a sparse matrix, spectral partitioning is still much slower than many other partitioning algorithms. Furthermore, the Lanczos algorithm becomes unstable as the Fiedler value approaches its neighboring eigenvalues (see [36, 30] for discussion of this problem). In fact, the eigenvector problem becomes fully degenerate if the Fiedler value assumes algebraic multiplicity greater than 1. For example, consider finding the Fiedler vector of a fully connected graph, for which the Fiedler value has algebraic multiplicity equal to  $n - 1$ . This situation could allow the Lanczos algorithm to converge to any vector in the subspace spanned by the eigenvectors corresponding to the Fiedler value.

In Figure 5.1, we compare the speed of the isoperimetric and spectral algorithms. Both partitions were computed using MATLAB, where the sparse, Lanczos-based, method for finding eigenvectors employs ARPACK [47] and solution to the system of linear equations is computed using the sparse linear solvers detailed in [28]. These experiments were done on a machine with a 2.39 GHz Intel Xeon chip and 3 GB of RAM. In order to illustrate the convergence problems for graphs having a Fiedler value with algebraic multiplicity greater than 1, we compared the two algorithms on a simple, square, unweighted, 4-connected lattice, for which the algebraic multiplicity of the Fiedler value is known to be 2. Graphs were tested with a number of nodes between  $9 \leq N \leq 50,000$ . Each algorithm is exhibited to have nearly linear behavior, although the constant is about an order of magnitude less for the isoperimetric algorithm when the algebraic multiplicity of the Fiedler value is greater than 1. Figure 5.1 also shows the experiments run with a graph composed of points placed randomly in the two-dimensional (2D) unit square with a uniform distribution and connected via a Delaunay triangulation. Here we still see an improvement for the speed of the isoperimetric algorithm (with a speed increase of about three times), but the behavior of the spectral runtimes illustrates the degeneracy problem—for those random graphs in which the second and third smallest eigenvalues are equal, or nearly equal, the running time of the spectral algorithm jumps to the plot obtained for the lattice. In other words, our experiments indicate that the isoperimetric algorithm operates

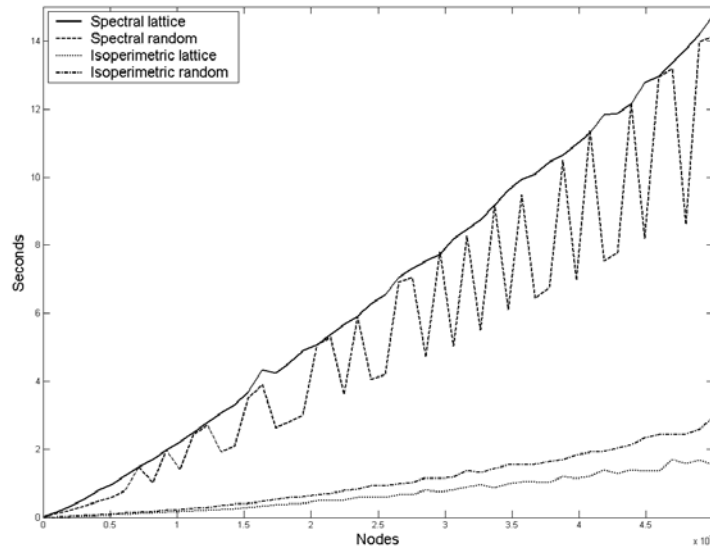


FIG. 5.1. Speed comparison of the spectral method to the isoperimetric algorithm for two classes of graphs: a square, 4-connected lattice, and a randomly distributed, planar, point set connected via Delaunay. See the text for details.

between three and ten times faster than the spectral algorithm, depending on the separation of the second and third eigenvalues. Additionally, the second experiment was run with just 30 randomly placed graphs with Delaunay connectivity. Therefore, the situations for which the degeneracy of the spectral approach is a concern do not appear to be rare. We note that the degeneracy problem of the spectral approach also precludes prediction of which runtime plot the spectral algorithm will employ (unless the spectrum is known a priori), while the isoperimetric algorithm has no such concerns. Finally, we note that the runtimes for the isoperimetric algorithm for each type of graph appear in Figure 5.1 to have a slightly different constant. We presume that this discrepancy is because the banded structure of the lattice Laplacian permits a simpler decomposition.

Finally, it has been pointed out [34] that a class of graphs exists for which spectral partitioning will produce consistently poor partitions.

**5.3. Geometric partitioning.** Geometric partitioning [29] is only defined for graphs with nodal coordinates specified (e.g., finite-element meshes). Furthermore, the geometric partitioning algorithm assumes that the nodes are locally connected and computes a good *spatial* separator (i.e., ignoring topology altogether). Building on theoretical results of [53, 52], geometric partitioning works by stereographically projecting nodes from the plane to the Riemann sphere, conformally mapping the nodes on the sphere such that a special point (called the *centerpoint*) is at the origin, and then randomly choosing any great circle on the Riemann sphere to divide the points into two equal halves. In practice, several great circles are randomly chosen and the best one is used as the output.

Although geometric partitioning is computationally inexpensive (albeit using multiple trials) and produces good partitions, the major drawback of the geometric partitioning algorithm is its inapplicability to graphs without coordinates or to graphs that are not locally connected (e.g., nonplanar graphs).

**5.4. Geometric-spectral partitioning.** Geometric-spectral partitioning [14] combines elements of both spectral partitioning and geometric partitioning. By finding the eigenvectors corresponding to both the second and third smallest eigenvalues, one may treat the nodes as having *spectral coordinates* by viewing the values associated with each node in the two eigenvectors as 2D coordinates in the plane. Applying geometric partitioning to the spectral coordinates of the nodes instead of actual coordinates (if available) heuristically gives better partitions than either spectral or geometric partitioning alone.

Geometric-spectral partitioning represents a more general algorithm than straightforward geometric partitioning since it applies to more general graphs (i.e., graphs without coordinates) and because it takes advantage of topological information in computing the spectral coordinates. However, geometric-spectral partitioning is more computationally expensive than spectral partitioning since it requires the computation of two eigenvectors instead of one. Numerically, the algorithm is also prone to more numerical problems than spectral partitioning, since the Lanczos algorithm has increased error as more “interior” eigenvectors are computed [30], and because the spectral coordinates are not unique if either the second or the third eigenvalue of  $L$  has an algebraic multiplicity greater than one.

**5.5. Multilevel Kernighan–Lin.** The Kernighan–Lin algorithm [41] employs a greedy approach for graph partitioning. Although fast, the final solution is highly dependent on the initial (random) partition used to speed the algorithm. Due to the speed, this algorithm may be run multiple times in order to achieve an acceptable partition.

By far the most successful use of the Kernighan–Lin algorithm is as a *refinement* technique. A multilevel representation of a graph may be obtained through the use of a maximal independent set [66]. If the coarsest level of the graph is partitioned through some means (e.g., spectral, geometric), then the quality of the partition may be refined to the next level by employing Kernighan–Lin. This multilevel Kernighan–Lin approach is the backbone for state-of-the-art graph partitioning software packages such as Chaco [36] and Metis [40]. We include the Metis [40] package in the comparison section.

One expects these algorithms to perform the best since they take into account multiple levels of graph representation. However, multilevel Kernighan–Lin may be easily combined with other graph partitioning techniques by employing the secondary technique to perform the partitioning at the coarsest level. Independent of the multilevel scheme, Kernighan–Lin may be used to refine a partition given by another algorithm.

One problem with the Kernighan–Lin approach is that the size of the partitions must be specified. This is not an issue for the graph bisection problem, where the partition size is required to be half of the node set. However, if one is interested in finding partitions close to the isoperimetric sets, then the algorithm must be flexible enough to find the best partitions of arbitrary size.

**6. Algorithm behavior.** As with spectral methods, the solution of (2.15) (or (3.7)) yields a continuous-valued result that must then be converted into a bipartition by thresholding. In this section we introduce two examples that illustrate the distribution of the potentials and their relationship to the ground node.

Figure 6.1 shows the solution to the spectral problem and (3.7) on a simple, square, 4-connected grid. Note that the square grid causes the degeneracy of the Fiedler vector required by the spectral approach, since the algebraic multiplicity of the Fiedler value is 2. Consequently, even for this simple graph, the Lanczos method

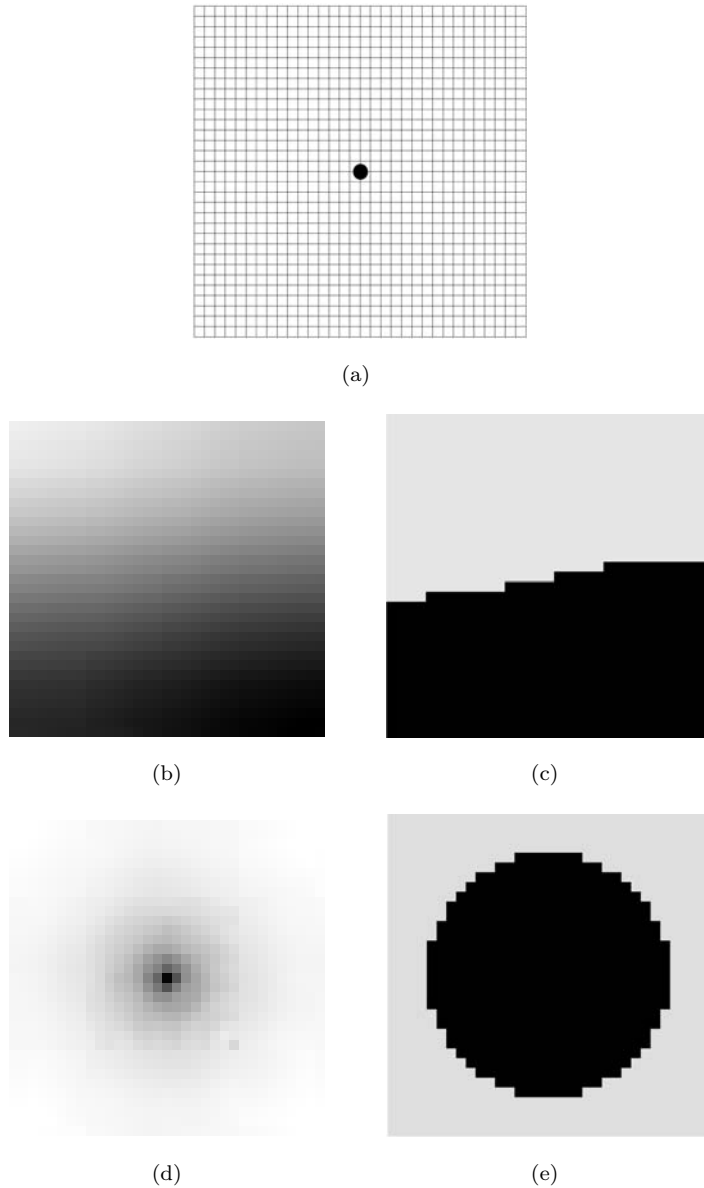


FIG. 6.1. Comparison of spectral and isoperimetric solutions on a square, unweighted, 4-connected lattice. (a) Original lattice. The black dot represents the location of the ground point for the isoperimetric algorithm. (b) Spectral solution. Note that the algebraic multiplicity of the Fiedler vector results in a degenerate solution—any vector in the subspace spanned by the Fiedler vectors is a valid output of the Lanczos algorithm. (c) Cut obtained from the spectral solution. (d) Solution of the isoperimetric algorithm, given a ground point in the lattice center. (e) Cut obtained from the isoperimetric solution. Despite the Manhattan distance imposed by 4-connectivity, note the similarity with the well-known solution to the isoperimetric problem in the Euclidean plane—a circle.

takes longer to converge and returns a vector in the subspace spanned by the multiple Fiedler vectors (as seen in Figure 5.1). In contrast, given a ground node in the center of the lattice, the isoperimetric algorithm returns a unique solution. The fact that our algorithm returns a “circle” on this graph lends justification to the name

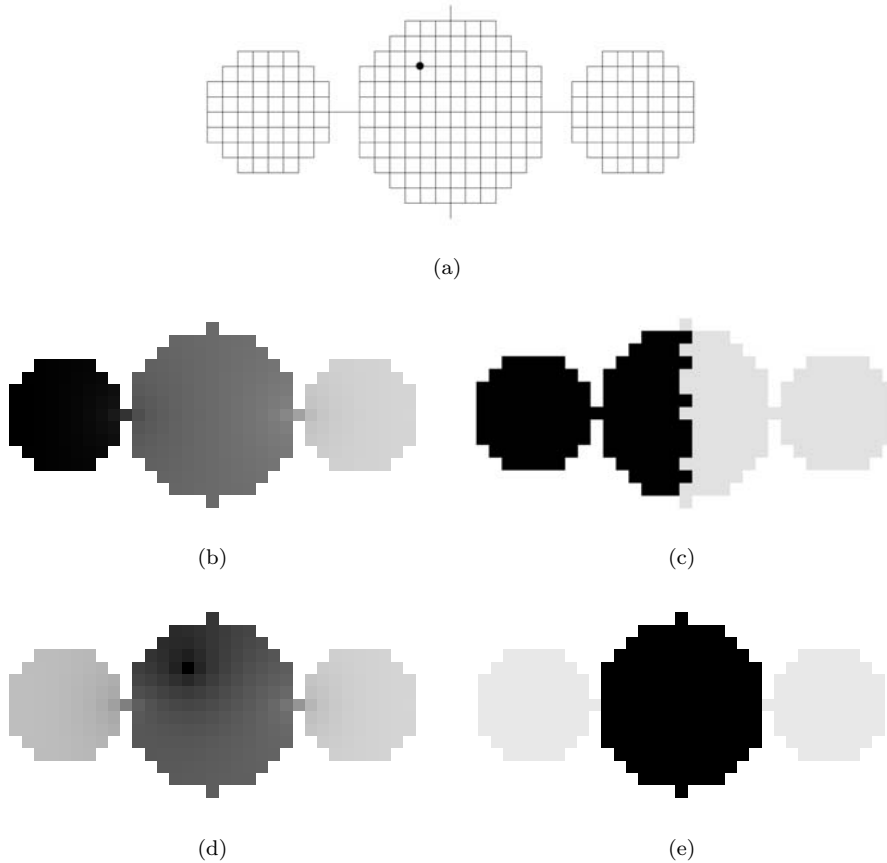


FIG. 6.2. Comparison of spectral and isoperimetric solutions on a graph comprises a large, central, “circle” with smaller, side “circles.” This graph is intended to build the reader’s intuition about the solution. (a) Original graph. (b) Spectral. The symmetry of the graph results in a symmetric solution over the central axis of symmetry. (c) Cut obtained from the spectral solution. Since the spectral solution is symmetric over the central axis of symmetry, the equal partition cut must also lie along this axis. For graphs of this type, such a solution may be arbitrarily poor. (d) Solution of the isoperimetric algorithm, given a ground point in the central circle, designated by the black dot. Note that the ground is offset from the circle center. (e) Cut obtained from the isoperimetric solution. Note that the cut is not simply a circle (relative to the graph distance), as otherwise the partition would tilt toward the side circle closer to the ground, which it does not.

“isoperimetric” algorithm, since the circle has been known since antiquity to be the solution to the isoperimetric problem in the Euclidean plane [56] and a lattice is widely used as a discrete approximation to the plane when solving PDEs, etc. This appearance of the circle should be no surprise, given the motivation of the algorithm or the random walks interpretation, since it is well known in the Euclidean plane that one is expected to stand on a circle with radius  $\sqrt{n}$  after taking  $n$  random steps from a central point [49] (given here by the grounded node). However, we stress that the “circle” partition in Figure 6.1 is *not* simply the locus of points within a certain distance from the ground, since the 4-connectivity of the lattice would induce a diamond-shaped locus.

Figure 6.2 shows another simple graph, constructed by windowing a 4-connected lattice in the shape of a large, central “circle,” weakly connected to two side “circles,”

such that the cardinality of the central circle equals the sum of the side circles. The symmetry of this graph across an unfavorable cut axis (i.e., the center of the large circle) leads to problems for the spectral algorithm, as well illustrated in Guattery and Miller’s “roach” graph [34]. In contrast, the isoperimetric algorithm cuts the side circles off the central circle, even though the ground is offset from the center of the circle. Again, this should be no surprise given the electrical circuit analogy—all of the “current” entering nodes in the side circles must flow through the connecting branch, yielding a huge voltage drop across this edge. Consequently, the center circle is widely separated from the side circles. These experiments demonstrate that the “background” partition (i.e.,  $\bar{S}$ ) may or may not be connected. As with the above experiment, the partition is not simply the locus of all points equidistant from the ground. The ground point was offset from the center of the large circle in Figure 6.2, but the partition boundary does not “leak” into the closer side circle.

**7. Comparison of algorithms.** The graph partition quality produced by the algorithms reviewed above was examined on weighted and unweighted planar, non-planar and three-dimensional (3D) graphs, as well as some specialized graphs. All algorithms were compared by considering the number of edges spanning the two equally sized partitions produced by each algorithm (i.e., the median cut, for the isoperimetric and spectral algorithms). The spectral and isoperimetric algorithms were additionally compared in terms of the isoperimetric ratio by employing the criterion cut to find the best partition (i.e., not necessarily partitions of equal size). Note that the “uniform” volume definition, i.e., (2.5), was used to evaluate the isoperimetric ratio for these algorithms, since this is the definition most commonly desired in graph partitioning problems. As was noted in [60] we found that the distribution of spanning edges and  $h(x)$  resembled Gaussian distributions. For this reason, we report the mean and variance of the number of edges spanning the partitions (or  $h(x)$ ) for the 1,000 graphs generated of each type.

Four versions of the isoperimetric algorithm are compared with the algorithms reviewed above on the graph partitioning problem. The four versions are obtained from the two notions of volume corresponding to (2.15) and (3.7) and two grounding strategies: grounding the maximum degree node and grounding a random node. In the following tables, grounding strategies are denoted by “MG” for maximum degree ground and “RG” for random ground. Volume definitions are designed by “Uni” for the cardinality definition of volume of (2.15) or “Deg” for the degree-based definition of volume given by solution to (3.7). For each random ground partition, three randomly chosen grounds were tried and the best one chosen. The geometric, and geometric-spectral, partitions were all obtained through the MESHPART software package for MATLAB [29] and Metis was executed using the Metis package of [40].

Relative runtimes of the algorithms were considered for a 10,000-node, randomly connected, planar graph of the type described in section 7.1. From fastest to slowest were Metis (0.031 s), Isoperimetric (0.329 s), Geometric (0.500 s), Spectral (2.125 s), and Geo-Spectral (6.531 s). However, as was mentioned above, the code to execute each of these algorithms was written by different groups, with varying dependence on MATLAB.

**7.1. Planar graphs.** Planar graphs were generated by uniformly sampling 1,000 points from a 2D unit square and connecting them with a Delaunay triangulation. One thousand such graphs were generated for both the weighted and unweighted trials. In the weighted trial, weights were randomly assigned to each edge from a uniform distribution on the interval  $[0, 1 \times 10^4]$ . Results for the median cut comparison are

TABLE 7.1

Comparison of the algorithms on 1,000 randomly generated planar graphs produced by uniformly sampling 1,000 2D points in the unit square and connecting via a Delaunay triangulation (see section 7.1). The two leftmost columns represent the mean and variance of  $|\partial S|$  for equal-sized partitions produced by each algorithm when applied to unweighted planar graphs. The two rightmost columns represent the same quantities for weighted planar graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean cut	Variance cut	Mean cut	Variance cut
Iso MG Uni	80.8	116	$3.76 \times 10^5$	$3.69 \times 10^9$
Iso MG Deg	80.7	117	$3.75 \times 10^5$	$3.61 \times 10^9$
Iso RG Uni	77.9	44.9	$3.51 \times 10^5$	$1.5 \times 10^9$
Iso RG Deg	78.1	43	$3.5 \times 10^5$	$1.27 \times 10^9$
Spectral	77.1	57.7	$3.47 \times 10^5$	$1.79 \times 10^9$
Geometric	70.6	12.1	$3.54 \times 10^5$	$7.94 \times 10^8$
Spectral geometric	66.8	9.81	$3.34 \times 10^5$	$7.97 \times 10^8$
Metis	66.9	23.5	$3.35 \times 10^5$	$1.15 \times 10^9$

TABLE 7.2

Comparison of the isoperimetric and spectral algorithms on 1,000 randomly generated planar graphs produced by uniformly sampling 1,000 2D points in the unit square and connecting via a Delaunay triangulation (see section 7.1). The two leftmost columns represent the mean and variance of  $h(x)$  obtained by applying the criterion cut to the output of each algorithm when applied to unweighted planar graphs. The two rightmost columns represent the same quantities for weighted planar graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean $h(x)$	Variance $h(x)$	Mean $h(x)$	Variance $h(x)$
Iso MG Uni	0.154	0.0004	683	$1.15 \times 10^4$
Iso MG Deg	0.154	0.000398	683	$1.17 \times 10^4$
Iso RG Uni	0.148	0.000131	635	$3.75 \times 10^3$
Iso RG Deg	0.148	0.000129	636	$3.7 \times 10^3$
Spectral	0.147	0.000201	625	$4.65 \times 10^3$

found in Table 7.1 and for the criterion cut comparison are found in Table 7.2.

**7.2. Nonplanar graphs.** Purely random graphs (typically nonplanar) were generated by connecting each pair of 1,000 nodes with a 1% probability. One thousand such graphs were generated for both the weighted and unweighted trials. In the weighted trial, weights were randomly assigned to each edge from a uniform distribution on the interval  $[0, 1 \times 10^4]$ . Since coordinate information was meaningless, only those algorithms which made use of topological information were compared (i.e., the isoperimetric, spectral, geometric-spectral, and Metis approaches). Results for the median cut comparison are found in Table 7.3 and for the criterion cut comparison in Table 7.4.

**7.3. 3D graphs.** Modeling and computer graphics applications frequently require the use of locally connected points in three dimensions. One thousand graphs were generated by uniformly sampling 1,000 points in the unit cube and connecting them via the 3D Delaunay. Both weighted and unweighted graphs were generated. In the weighted trial, weights were randomly assigned to each edge from a uniform distribution on the interval  $[0, 1 \times 10^4]$ . Results for the median cut comparison are found in Table 7.5 and for the criterion cut comparison in Table 7.6.



TABLE 7.3

Comparison of the algorithms on 1,000 random graphs (symmetric) produced by connecting each edge with a 1% probability (see section 7.2). Partitioning algorithms that rely on coordinate information were not included, since they are not applicable to this problem. The two leftmost columns represent the mean and variance of  $|\partial S|$  for an equal-sized bipartition produced by each algorithm for unweighted random graphs. The two rightmost columns represent the same quantities for weighted random graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean cut	Variance cut	Mean cut	Variance cut
Iso MG Uni	$4.47 \times 10^3$	$7.06 \times 10^3$	$2.14 \times 10^7$	$1.71 \times 10^{11}$
Iso MG Deg	$4.46 \times 10^3$	$5.62 \times 10^3$	$2.1 \times 10^7$	$1.83 \times 10^{11}$
Iso RG Uni	$4.14 \times 10^3$	$5.75 \times 10^3$	$2.11 \times 10^7$	$1.32 \times 10^{11}$
Iso RG Deg	$4.11 \times 10^3$	$6.46 \times 10^3$	$2.06 \times 10^7$	$1.41 \times 10^{11}$
Spectral	$4.06 \times 10^3$	$5.99 \times 10^3$	$2.05 \times 10^7$	$6.62 \times 10^{11}$
Spectral geometric	$3.98 \times 10^3$	$3.85 \times 10^3$	$1.99 \times 10^7$	$1.54 \times 10^{11}$
Metis	$3.43 \times 10^3$	$2.4 \times 10^3$	$1.72 \times 10^7$	$1.03 \times 10^{11}$

TABLE 7.4

Comparison of the algorithms on 1,000 random graphs (symmetric) produced by connecting each edge with a 1% probability (see section 7.2). The two leftmost columns represent the mean and variance of  $h(x)$  for a partition generated using the criterion cut for random graphs. The two rightmost columns represent the same quantities for weighted random graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean $h(x)$	Variance $h(x)$	Mean $h(x)$	Variance $h(x)$
Iso MG Uni	8.84	0.104	$4.09 \times 10^4$	$3.86 \times 10^6$
Iso MG Deg	8.92	0.0225	$4.22 \times 10^4$	$9.93 \times 10^5$
Iso RG Uni	8.22	0.0486	$3.91 \times 10^4$	$7.73 \times 10^6$
Iso RG Deg	8.19	0.0207	$4.1 \times 10^4$	$5.87 \times 10^5$
Spectral	8.12	0.0244	$3.98 \times 10^4$	$2.26 \times 10^6$

TABLE 7.5

Comparison of the algorithms on 1,000 randomly generated 3D produced by uniformly sampling 1,000 3D points in the unit cube and connecting via a 3D Delaunay (see section 7.3). The two leftmost columns represent the mean and variance of  $|\partial S|$  for an equal-sized bipartition produced by each algorithm for unweighted planar graphs. The two rightmost columns represent the same quantities for weighted planar graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean cut	Variance cut	Mean cut	Variance cut
Iso MG Uni	630	$2.14 \times 10^3$	$3.2 \times 10^6$	$7.61 \times 10^{10}$
Iso MG Deg	628	$2.12 \times 10^3$	$3.17 \times 10^6$	$7.39 \times 10^{10}$
Iso RG Uni	633	$1.17 \times 10^3$	$3.17 \times 10^6$	$3.27 \times 10^{10}$
Iso RG Deg	632	$1.16 \times 10^3$	$3.14 \times 10^6$	$3.24 \times 10^{10}$
Spectral	609	$1.31 \times 10^3$	$3.37 \times 10^6$	$9.05 \times 10^{11}$
Geometric	579	348	$2.9 \times 10^6$	$1.38 \times 10^{10}$
Spectral geometric	561	515	$2.81 \times 10^6$	$1.96 \times 10^{10}$
Metis	551	614	$2.75 \times 10^6$	$2.16 \times 10^{10}$

**7.4. Specialized graphs.** In addition to the randomly generated graphs used above to benchmark the isoperimetric algorithm, we also applied the set of algorithms to 2D graphs taken from applications (see [29, 14, 13] for other uses of these graphs).

TABLE 7.6

Comparison of the isoperimetric and spectral algorithms on 1,000 randomly generated 3D graphs produced by uniformly sampling 1,000 3D points in the unit cube and connecting via a 3D Delaunay (see section 7.3). The two leftmost columns represent the mean and variance of  $h(x)$  obtained using the criterion cut for each algorithm on unweighted planar graphs. The two rightmost columns represent the same quantities for weighted planar graphs.

Algorithm	Unweighted graphs		Weighted graphs	
	Mean $h(x)$	Variance $h(x)$	Mean $h(x)$	Variance $h(x)$
Iso MG Uni	1.26	0.0086	$6.39 \times 10^3$	$4.36 \times 10^5$
Iso MG Deg	1.25	0.00857	$6.54 \times 10^3$	$4.29 \times 10^5$
Iso RG Uni	1.25	0.00413	$6.1 \times 10^3$	$2.01 \times 10^5$
Iso RG Deg	1.25	0.00413	$6.29 \times 10^3$	$2.03 \times 10^5$
Spectral	1.22	0.00527	$7.13 \times 10^3$	$5.77 \times 10^6$

TABLE 7.7

Information about the specialized graphs used to benchmark the algorithms on. All graphs were obtained from the ftp site of John Gilbert and the Xerox Corporation at ftp.parc.xerox.com in the file /pub/gilbert/meshpart.uu.

Mesh Name	Nodes	Edges
Eppstein	547	1556
Tapir	1024	2846
“Crack”	136	354
Airfoil1	4253	12289
Airfoil2	4720	13722
Airfoil3	15606	45878
Spiral	1200	3191
Triangle	5050	14850

TABLE 7.8

The number of edges cut by the equal-sized partitions output by the various algorithms (see section 7.4). Note that since the Eppstein mesh is weighted, the cut cost is not an integer.

Algorithm	Epp	Tapir	Tri	Air1	Air2	Air3	Crack	Spiral
Iso MG Uni	24.8	51	150	137	111	255	200	9
Iso MG Deg	24.8	51	150	137	111	255	197	9
Iso RG Uni	20.2	33	158	105	123	188	152	9
Iso RG Deg	20.4	40	150	104	125	212	161	9
Spectral	21.6	58	152	132	117	194	157	9
Geometric	22.9	34	148	102	103	152	191	58
Geometric-spectral	22.4	23	144	88	104	159	149	9
Metis	27.3	24	144	85	96	141	142	9

The meshes were obtained through the ftp site of John Gilbert and the Xerox Corporation at ftp.parc.xerox.com from the file /pub/gilbert/meshpart.uu. The list of graphs used is given in Table 7.7. The results using the median cut comparison are found in Table 7.8 and for the criterion cut comparison are found in Table 7.9.

The set of algorithms were also applied to three graph families of theoretical interest. The first of these is the “roach” graph of [34] with the total length of the roach ranging from 10 to 50 nodes long (i.e., 20–100 nodes total). The family of roach graphs is known to result in poor partitions when spectral partitioning is employed with the median cut. For a roach with an equal number of “body” and “antennae” segments, the spectral algorithm will always produce a partition with  $|\partial S| = \Theta(n)$  (where  $\Theta(\cdot)$  is the function of [43]) instead of the constant cut set of two edges obtained

TABLE 7.9

The isoperimetric ratio obtained by applying the criterion cut method to the output of the isoperimetric and spectral partitioning algorithms (see section 7.4).

Algorithm	Epp	Tapir	Tri	Air1	Air2	Air3	Crack	Spiral
Iso MG Uni	0.0906	0.0474	0.0588	0.0351	0.0425	0.0268	0.0758	0.0108
Iso MG Deg	0.0894	0.0473	0.0587	0.0351	0.0425	0.0268	0.0759	0.0108
Iso RG Uni	0.0714	0.0435	0.0594	0.0329	0.0434	0.0212	0.057	0.0108
Iso RG Deg	0.0839	0.0286	0.0643	0.0329	0.043	0.0209	0.0572	0.0108
Spectral	0.0771	0.0262	0.0594	0.0342	0.0425	0.02	0.059	0.0108

TABLE 7.10

The mean number of edges cut by the equal-sized partitions output by the various algorithms over a parameter range for each family of graphs (see section 7.4). The three graph families here (roach, TreeXPath, and “badmesh”) are of theoretical interest in that they are known to produce poor results for different classes of partitioning algorithms.

Algorithm	Mean cut roach	Mean cut TreeXPath	Mean cut “badmesh”
Iso MG Uni	2	27.1	2.98
Iso MG Deg	2	26.7	2.98
Iso RG Uni	2.37	29.3	3.34
Iso RG Deg	2.32	27.8	3.1
Spectral	15.2	38	12.5
Geometric	2	26.4	453
Spectral geometric	2	27	2.98
Metis	2	27.5	2.98

TABLE 7.11

The mean isoperimetric ratio obtained using a criterion cut on the output of the partitioning algorithms when applied to three graph families of theoretical interest (see section 7.4). Means are calculated over a range of parameters defining the graph family (see the text for details).

Algorithm	Mean $h(x)$ roach	Mean $h(x)$ TreeXPath	Mean $h(x)$ “badmesh”
Iso MG Uni	0.0815	0.128	0.0297
Iso MG Deg	0.0815	0.128	0.0297
Iso RG Uni	0.0798	0.129	0.0294
Iso RG Deg	0.0798	0.127	0.0294
Spectral	0.0796	0.128	0.0294

by cutting the antennae from the body. It has been demonstrated [64] that the spectral approach may be made to correctly partition the roach graph if additional processing is performed. For this reason, the partitions obtained through use of the criterion cut are reasonable for the spectral algorithm. The second graph family of theoretical interest, referred to as “TreeXPath” [34], is known to result in poor partitions when spectral partitioning is used with the median cut. For purposes of benchmarking here, the cardinality of the 3D point set was varied between 50 and 1,000 nodes. The final graph family of theoretical interest is the so-called “badmesh” [13] for which no quality straight-line separator exists. Badmeshes were generated with node sets varying between 200 and 4,000, with a constant ratio of  $\frac{4}{5}$  between shell sizes. Mean values across the nodal range are reported in Table 7.10 for partitions obtained with the median cut and Table 7.11 using the criterion cut.

**8. Conclusion.** We have presented a new algorithm for partitioning graphs that requires the solution to a sparse, symmetric, positive-definite system of equations.

Analogous to Fiedler’s examination of the connectivity behavior of the spectral algorithm [25], we examined the connectivity of the partitions returned by the isoperimetric algorithm and proved that the partition containing the ground node must be connected. Formal behavior of the algorithm was also examined on fully connected graphs and trees. Empirically, the solution to the system of equations was examined on two graphs and interpreted in the context of two physical interpretations of the equations. The partition returned by application of the isoperimetric algorithm to a 4-connected, 2D lattice was shown to resemble the known solution to the isoperimetric problem on the 2D Euclidean plane—a circle.

Based on the above trials, it appears that a random ground approach generally works better than grounding the node with maximum degree. Therefore, we suggest the strategy of grounding the node with maximum degree and, if possible, additionally trying random grounds in order to choose the best partition. Although the difference in partition quality between the two definitions of combinatorial volume is slight, it seems the definition of (3.6) fared better over that of (2.5) for the equally sized partition, yet favored the cardinality-based volume measure in computing the partition with minimum isoperimetric ratio. However, the latter should not be surprising, since the isoperimetric ratio was computed using the cardinality-based volume measure. In practice, neither notion of volume suggests recommendation over the other for an equally sized cut. However, if trying to compute the partition with minimum isoperimetric ratio (i.e., the quotient cut), one should employ the appropriate notion of volume.

While competitive with the algorithms compared in this work, we acknowledge that the isoperimetric algorithm appears to give slightly higher averages than the other algorithms, most notably the multilevel KL approach. However, the isoperimetric algorithm has the following qualities not shared by all of the comparison algorithms:

1. Coordinate information is not needed, allowing application on abstract graphs.
2. Edge weights are taken into account when computing the partitions.
3. The algorithm returns a family of partitions (corresponding to different threshold choices) that allows a partition with the “best” cardinality to be chosen, instead of prespecifying the partition cardinality.

In fact, the only algorithm included here that shares these three properties is spectral partitioning, which is shown above to produce partitions of similar quality, but more slowly and at the risk of a degenerate solution. Furthermore, the isoperimetric algorithm appears to perform better (relative to the other algorithms) on the specific benchmark graphs than on the experimental random graphs. Therefore, it is possible that the family of random graphs used in the experiments may be unfavorable to the algorithm. In contrast, the experiments shown on the “roach,” “TreeXPath,” and “badmesh” graph families, which were designed to be problematic for other algorithms, showed little problem for the isoperimetric algorithm. We also note that some of the better-performing algorithms (e.g., spectral geometric) use the spectral approach as a component. Therefore, hybridization with the isoperimetric algorithm instead of the spectral approach might improve speed/stability issues associated with the spectral component of such an approach.

The isoperimetric algorithm for graph partitioning is implemented in the publicly available graph analysis toolbox [31, 32] for MATLAB obtainable at <http://eslab.bu.edu/software/graphanalysis/>. This toolbox was used to generate all of the results compiled in the tables of this paper. Further work with this algorithm might include

the development of a more principled grounding strategy and hybridization with other partitioning methods.

**Acknowledgment.** The authors would like to thank Jonathan Polimeni for many fruitful discussions and suggestions.

## REFERENCES

- [1] L. AHLFORS, *Complex Analysis*, McGraw-Hill, New York, 1966.
- [2] N. ALON, *Eigenvalues and expanders*, *Combinatorica*, 6 (1986), pp. 83–96.
- [3] N. ALON AND V. MILMAN,  $\lambda_1$ , *isoperimetric inequalities for graphs and superconcentrators*, *J. Combin. Theory Ser. B*, 38 (1985), pp. 73–88.
- [4] C. J. ALPERT AND A. B. KAHNG, *Recent directions in netlist partitioning: A survey*, *Integration: The VLSI J.*, 19 (1995), pp. 1–81.
- [5] W. N. ANDERSON, JR. AND T. D. MORLEY, *Eigenvalues of the Laplacian of a graph*, *Linear Multilinear Algebra*, 18 (1985), pp. 141–145.
- [6] G. ARFKEN AND H.-J. WEBER, EDs., *Mathematical Methods for Physicists*, 3rd ed., Academic Press, New York, 1985.
- [7] D. A. V. BAAK, *Variational alternatives of Kirchhoff's loop theorem in DC circuits*, *Amer. J. Phys.*, 67 (1998), pp. 36–44.
- [8] N. BIGGS, *Algebraic Graph Theory*, Cambridge Tracts in Math. 67, Cambridge University Press, Cambridge, UK, 1974.
- [9] J. BOLZ, I. FARMER, E. GRINSPUN, AND P. SCHRÖDER, *Sparse matrix solvers on the GPU: Conjugate gradients and multigrid*, *ACM Trans. Graphics*, 22 (2003), pp. 917–924.
- [10] F. H. BRANIN, JR., *The inverse of the incidence matrix of a tree and the formulation of the algebraic-first-order differential equations of an RLC network*, *IEEE Trans. Circuit Theory*, 10 (1963), pp. 543–544.
- [11] F. H. BRANIN, JR., *The algebraic-topological basis for network analogies and the vector calculus*, in *Proceedings of the Symposium on Generalized Networks*, J. Fox, ed., Polytechnic Press, Brooklyn, NY, 1966, pp. 453–491.
- [12] T. N. BUI AND C. JONES, *A heuristic for reducing fill-in in sparse matrix factorization*, in *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, Vol. 1, R. F. Sincovec, D. Keyes, M. Leuze, L. Petzold, and D. Reed, eds., SIAM, Philadelphia, 1993, pp. 445–452.
- [13] F. CAO, J. R. GILBERT, AND S.-H. TENG, *Partitioning Meshes with Lines and Planes*, Technical report CSL-96-01, Palo Alto Research Center, Xerox Corporation, 1996.
- [14] T. F. CHAN, J. R. GILBERT, AND S.-H. TENG, *Geometric Spectral Partitioning*, Technical report CSL-94-15, Palo Alto Research Center, Xerox Corporation, 1994.
- [15] J. CHEEGER, *A lower bound for the smallest eigenvalue of the Laplacian*, in *Problems in Analysis*, R. Gunning, ed., Princeton University Press, Princeton, NJ, 1970, pp. 195–199.
- [16] F. R. K. CHUNG, *Spectral Graph Theory*, CBMS Reg. Conf. Ser. Math. 92, AMS, Providence, RI, 1997.
- [17] J. DODZIUK, *Difference equations, isoperimetric inequality and the transience of certain random walks*, *Trans. Am. Math. Soc.*, 284 (1984), pp. 787–794.
- [18] J. DODZIUK AND W. S. KENDALL, *Combinatorial Laplacians and the isoperimetric inequality*, in *From Local Times to Global Geometry, Control and Physics*, Pitman Res. Notes Math. Ser. 150, K. D. Ellworthy, ed., Longman Scientific and Technical, London, 1986, pp. 68–74.
- [19] W. DONATH AND A. HOFFMAN, *Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices*, *IBM Tech. Disclosure Bull.*, 15 (1972), pp. 938–944.
- [20] J. J. DONGARRA, I. S. DUFF, D. C. SORENSON, AND H. A. VAN DER VORST, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, 1991.
- [21] P. DOYLE AND L. SNELL, *Random Walks and Electric Networks*, Carus Math. Monogr. 22, Mathematical Association of America, Washington, D.C., 1984.
- [22] C. M. FIDUCCIA AND R. M. MATTHEYES, *A linear-time heuristic for improving network partitions*, in *Proceedings of the 19th ACM/IEEE Design Automation Conference*, Las Vegas, 1982, IEEE Press, Piscataway, NJ, pp. 175–181.
- [23] M. FIEDLER, *Algebraic connectivity of graphs*, *Czechoslovak. Math. J.*, 23 (1973), pp. 298–305.
- [24] M. FIEDLER, *Eigenvectors of acyclic matrices*, *Czechoslovak. Math. J.*, 25 (1975), pp. 607–618.
- [25] M. FIEDLER, *A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory*, *Czechoslovak. Math. J.*, 25 (1975), pp. 619–633.

- [26] M. FIEDLER, *Special Matrices and their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, Dordrecht, The Netherlands, 1986.
- [27] L. FOULDS, *Graph Theory Applications*, Universitext, Springer-Verlag, New York, 1992.
- [28] J. GILBERT, C. MOLER, AND R. SCHREIBER, *Sparse matrices in MATLAB: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.
- [29] J. R. GILBERT, G. L. MILLER, AND S.-H. TENG, *Geometric mesh partitioning: Implementation and experiments*, SIAM J. Sci. Comput., 19 (1998), pp. 2091–2110.
- [30] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [31] L. GRADY, *Space-Variant Computer Vision: A Graph-Theoretic Approach*, Ph.D. thesis, Boston University, Boston, MA, 2004.
- [32] L. GRADY AND E. L. SCHWARTZ, *The Graph Analysis Toolbox: Image Processing on Arbitrary Graphs*, Technical report TR-03-021, Boston University, Boston, MA, 2003.
- [33] K. GREMBAN, *Combinatorial Preconditioners for Sparse, Symmetric Diagonally Dominant Linear Systems*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.
- [34] S. GUATTEY AND G. MILLER, *On the quality of spectral separators*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 701–719.
- [35] K. M. HALL, *An  $r$ -dimensional quadratic placement algorithm*, Manag. Sci., 17 (1970), pp. 219–229.
- [36] B. HENDRICKSON AND R. LELAND, *The Chaco User's Guide*, Technical report SAND95-2344, Sandia National Laboratory, Albuquerque, NM, 1995.
- [37] B. HENDRICKSON AND R. LELAND, *A multilevel algorithm for partitioning graphs*, in Proceedings of the 1995 ACM/IEEE Conference on Supercomputing, San Diego, CA, S. Karin, ed., ACM Press, New York, 1995.
- [38] B. HENDRICKSON, R. LELAND, AND R. VAN DRIESSCHE, *Skewed graph partitioning*, in Proceedings of the Eighth SIAM Conference on Parallel Processing, M. Heath, V. Torczon, G. Astfalk, P. E. Bjorstad, A. H. Karp, C. H. Koebel, V. Kumar, R. F. Lucas, L. T. Watson, and D. E. Womble, eds., SIAM, Philadelphia, 1997.
- [39] Y. HU AND R. BLAKE, *Numerical experiences with partitioning of unstructured meshes*, Parallel Comput., 20 (1994), pp. 815–829.
- [40] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1998), pp. 359–393.
- [41] B. KERNIGHAN AND S. LIN, *An efficient heuristic procedure for partitioning graphs*, Bell Syst. Tech. J., 49 (1970), pp. 291–308.
- [42] J. M. KLEINHAUS, G. SIGL, F. M. JOHANNES, AND K. J. ANTREICH, *GORDIAN: VLSI placement by quadratic programming and slicing optimization*, IEEE Trans. Comput. Aided Design, 10 (1991), pp. 356–365.
- [43] D. E. KNUTH, *Big omicron and big omega and big theta*, SIGACT News, 8 (1976), pp. 18–24.
- [44] U. R. KODRES, *Geometrical positioning of circuit elements in a computer*, in Proceedings of the 1959 AIEE Fall General Meeting, CP59-1172, Chicago, IL, AIEE Press, New York, 1959.
- [45] U. R. KODRES, *Partitioning and card selection*, in Design Automation of Digital Systems, Vol. 1, M. A. Breuer, ed., Prentice-Hall, Englewood Cliffs, NJ, 1972, Chap. 4, pp. 173–212.
- [46] M. KUNZ, *A Möbius inversion of the Ulam subgraphs conjecture*, J. Math. Chem., 9 (1992), pp. 297–305.
- [47] R. B. LEHOUCQ, D. C. SORENSON, AND C. YANG, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
- [48] T. LEIGHTON AND S. RAO, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM, 46 (1999), pp. 787–832.
- [49] J. D. LOGAN, *An introduction to nonlinear partial differential equations*, Pure Appl. Math. (N. Y.), John Wiley, New York, 1994.
- [50] J. C. MAXWELL, *A Treatise on Electricity and Magnetism*, 3rd ed., Vol. 1, Dover, New York, 1991.
- [51] R. MERRIS, *Laplacian matrices of graphs: A survey*, Linear Algebra Appl., 197–198 (1994), pp. 143–176.
- [52] G. L. MILLER, S.-H. TENG, W. P. THURSTON, AND S. A. VAVASIS, *Automatic mesh partitioning*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., IMA Vol. Math. Appl. 56, Springer-Verlag, Berlin, 1993, pp. 57–84.
- [53] G. L. MILLER, S.-H. TENG, W. P. THURSTON, AND S. A. VAVASIS, *Geometric separators for finite-element meshes*, SIAM J. Sci. Comput., 19 (1998), pp. 364–386.
- [54] B. MOHAR, *Isoperimetric inequalities, growth and the spectrum of graphs*, Linear Algebra Appl., 103 (1988), pp. 119–131.

- [55] B. MOHAR, *Isoperimetric numbers of graphs*, J. Combin. Theory Ser. B, 47 (1989), pp. 274–291.
- [56] G. PÓLYA AND G. SZEGÖ, *Isoperimetric Inequalities in Mathematical Physics*, Ann. of Math. Stud. 56, Princeton University Press, Princeton, NJ, 1951.
- [57] A. POTHEN, H. SIMON, AND K.-P. LIOU, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.
- [58] A. POTHEN, H. SIMON, AND L. WANG, *Spectral Nested Dissection*, Technical report CS-92-01, Pennsylvania State University, 1992.
- [59] B. M. RIESS, K. DOLL, AND F. M. JOHANNES, *Partitioning very large circuits using analytical placement techniques*, in Proceedings of the 31st ACM/IEEE Design Automation Conference, IEEE Press, Piscataway, NJ, 1994, pp. 646–651.
- [60] G. R. SCHREIBER AND O. C. MARTIN, *Cut size statistics of graph bisection heuristics*, SIAM J. Optim., 10 (1999), pp. 231–251.
- [61] J. SHI AND J. MALIK, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intelligence, 22 (2000), pp. 888–905.
- [62] G. SIGL, K. DOLL, AND F. M. JOHANNES, *Analytical placement: A linear of a quadratic objective function?*, in Proceedings of the 28th ACM/IEEE Design Automation Conference, IEEE Press, Piscataway, NJ, 1991, pp. 427–432.
- [63] H. D. SIMON, *Partitioning of unstructured problems for parallel processing*, Comput. Syst. Engrg., 2 (1991), pp. 135–148.
- [64] D. A. SPIELMAN AND S.-H. TENG, *Spectral Partitioning Works: Planar Graphs and Finite Element Meshes*, Technical report UCB CSD-96-898, University of California, Berkeley, 1996.
- [65] G. STRANG, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.
- [66] S.-H. TENG, *Coarsening, Sampling and Smoothing: Elements of the Multilevel Method*, manuscript, 1997.
- [67] P. TETALI, *Random walks and the effective resistance of networks*, J. Theor. Probab., 4 (1991), pp. 101–109.
- [68] Z. WU AND R. LEAHY, *An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation*, IEEE Pattern Anal. Mach. Intelligence, 11 (1993), pp. 1101–1113.