Spectral K-Way Ratio-Cut Partitioning and Clustering

Pak K. Chan, *Member, IEEE,* Martine D. F. Schlag, *Member, IEEE,* and Jason Y. Zien

***Abstract*—Recent research on partitioning has focused on the ratio-cut cost metric, which maintains a balance between the cost of the edges cut and the sizes of the partitions without fixing the size of the partitions *a priori*. Iterative approaches and spectral approaches to two-way ratio-cut partitioning have yielded higher quality partitioning results. In this paper, we develop a spectral approach to multi-way ratio-cut partitioning that provides a generalization of the ratio-cut cost metric to k-way partitioning and a lower bound on this cost metric. Our approach involves finding the k smallest eigenvalue/eigenvector pairs of the Laplacian of the graph. The eigenvectors provide an embedding of the graph's n vertices into a k-dimensional subspace. We devise a time and space efficient clustering heuristic to coerce the points in the embedding into k partitions. Advancement over the current work is evidenced by the results of experiments on the standard benchmarks.**

***Index Terms*— Partitioning, multiway partitioning, ratio-cut metric, spectral method, clustering, eigenvalues, Lanczos algorithm.**

I. Introduction

Graph partitioning is a fundamental problem in the field of design automation. The goal of graph partitioning is to divide the vertices of a graph with weighted edges into several subsets subject to size or balance constraints while minimizing the interconnections among those subgraphs. Among the many uses for graph partitioning are: placement of circuits [1], subdividing a circuit for packaging into separate components with the goal of minimizing the pin count of the modules or the delay, and subdividing a circuit into components for parallel logic simulation with the goal of minimizing the communication [2]. An emerging area in which graph partitioning will play an important role is in rapid prototyping with multiple FPGA's. As circuits grow larger, efficient and effective graph partitioning algorithms become a more crucial part of the design process.

Without size balancing constraints, graph partitioning can be solved optimally in polynomial time by network flow (max-flow/min-cut) algorithms. Unfortunately, solutions to the network flow problem tend to yield extremely unbalanced partitions [2], and when size balancing constraints are imposed, the problem becomes NP-complete. Because of its importance, many heuristics have been devised to solve the graph partitioning problem. These heuristics may be classified as:

1) *Hypergraph-based versus graph-based*—In a hypergraph, edges may have degree greater than two. Multi-pin nets of circuits are very well-represented by the hypergraph model. In a graph, edges all have degree two. When the original problem is presented as a hypergraph, *net models* can be used to approximate the hypergraph by a graph.
2) *Spectral versus iterative*—In *spectral partitioning* techniques, the eigenvectors and eigenvalues (spectrum) of a graph are computed and a cost function is shown to be minimized by a function of the spectrum. Some heuristic for mapping the information provided by the eigenvectors into an actual partition is devised. With the advent of high-performance workstations with floating-point accelerators, this avenue of partitioning has become practical. *Iterative* heuristics explore the solution space by making a large number of moves (small changes to the solution) either randomly or greedily in an attempt to discover a global minimum [3], [4].
3) *Multi-way (k-way) versus two-way*—A k-way partitioning divides the hypergraph into k vertex disjoint subgraphs [5], whereas bipartitioning (two-way) divides a hypergraph into exactly two subgraphs [6]. Any two-way method can be applied recursively to generate k partitions, but the quality of the solution may suffer.

Recently, Leighton and Rao [7] and Wei and Cheng [2] have shown that the quality of 2-way partitions can be improved by incorporating the balancing of partition sizes in the cost metric, rather than imposing constraints on the partition sizes. They define the ratio-cut cost metric for a 2-way partition to be $E_c/|P_1||P_2|$ where E_c is sum of the weights of edges cut and $|P_1|$ and $|P_2|$ are the sizes of the two partitions [7], [8].

This paper describes a new k-way, spectral, graph-based partitioning method. Three main results on partitioning are presented that reveal the nature of the relation between ratio-cut partitioning and spectral-based partitioning:

1) a generalization of the 2-way ratio-cut cost metric to a k-way ratio-cut cost metric;
2) an efficient heuristic to effect a k-way partitioning based on the k-dimensional spectral embedding of a graph; and
3) a lower bound on the optimal k-way ratio-cut partitioning.

Manuscript received October 18, 1993; revised March 11, 1994. This work was supported in part by NSF Grants MIP-9196276 and MIP-9223740 and the MICRO Program of the University of California (P. K. Chan) and by NSF Grants MIP-8896276 and MIP-9223740 (M. D. F. Schlag). This is an extended version of a conference paper that appeared in the 30th Design Automation Conference, Dallas, ACM/IEEE, pp. 749–754, June, 1993. This paper was recommended by Associate Editor G. Zimmermann.

The authors are with the Computer Engineering Department, University of California-Santa Cruz, Santa Cruz, CA 95064 USA.

IEEE Log Number 9401859.

II. Problem Formulation

An instance of the graph-partitioning problem consists of a graph, $G = (\mathcal{V}, \mathcal{E})$ with vertices, $\mathcal{V} = \{\nu_1, \nu_2, \cdots, \nu_n\}$, and weighted edges where the weight of edge $e = (\nu_i, \nu_j)$ represents the cost of putting ν_i and ν_j in separate partitions. The problem is to find a partition of the set of vertices $\mathcal{P} = \{P_1, P_2, P_3, \cdots P_k\}$ for a given k, which optimizes some cost criterion based on the weights of the edges cut and/or the sizes of the partitions.

Iterative-based partitioning heuristics are local in nature while spectral-based partitioning extracts global information about the structure of a graph from the eigenvalues/eigenvectors for graph matrices. The relation between the properties of a graph and its *spectrum* (the eigenvalues/eigenvectors of its associated matrices) has been an area of active research for several years [9]–[12]. The spectra of the adjacency matrix and Laplacian of a graph are the basis for both partitioning [13]–[15] and placement techniques [16]. The *adjacency matrix* of G is the $n \times n$ matrix $A(G) = [a_{ij}]$ where a_{ij} is the weight of the edge between ν_i and ν_j. The *degree matrix* of G is the $n \times n$ matrix $D(G) = [d_{ij}]$ defined by

$$d_{ij} = \begin{cases} \displaystyle\sum_{k=1}^{n} a_{ik} & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

The *Laplacian* of G is the $n \times n$ symmetric matrix $Q(G) = D(G) - A(G)$. Since the rows (and columns) sum to 0, the Laplacian is singular; it has rank at most $n - 1$ and 0 as an eigenvalue. In fact, the multiplicity of the 0 eigenvalue is the number of connected components of G [10]. Fig. 1 shows a simple 14-node graph. Assuming unit-cost edges its Laplacian Q would be the matrix shown at the bottom of the page.

Donath and Hoffman [14] derived a lower bound on the weight of the edges cut (E_c) by a partition satisfying predetermined partition sizes. If $m_1 \geq m_2 \geq \cdots \geq m_k$ are the given partition sizes and $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k$ are the smallest k eigenvalues of the Laplacian, then $E_c \geq 1/2\, \Sigma_{i=1}^{k} \lambda_i m_i$. Later, Boppana [9], and Rendl/Wolkowicz [17] improve Donath and Hoffman's bounds. Barnes [13] formulates the k-way partitioning problem in terms of an *assignment* matrix. A k-way partition of the vertices of $G, \mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ is represented by an $n \times k$ *assignment matrix* $Y(\mathcal{P}) = [y_{ih}]$ where

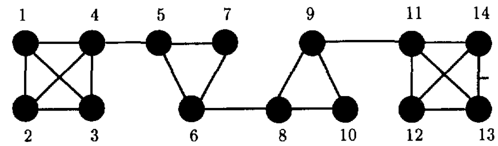


Fig. 1. A simple 14-node graph.

$$y_{ih} = \begin{cases} 1 & \text{if } \nu_i \in P_h \\ 0 & \text{if } \nu_i \notin P_h. \end{cases} \tag{1}$$

A k-partition of $G, \mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ can also be represented by the $n \times n$ *partition matrix* [13], $P = [p_{ij}]$ where

$$p_{ij} = \begin{cases} 1 & \text{if } \nu_i \text{ and } \nu_j \text{ are in the same partition} \\ 0 & \text{otherwise.} \end{cases}$$

For a given partition $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$, we have that $P = YY^T$. Given pre-determined partition sizes, $m_1, \cdots, m_k$, Barnes solves the partitioning problem by finding an approximation of the adjacency matrix A by a partition matrix P with respect to the Frobenius norm. Barnes's method involves finding the first k eigenvectors of A followed by solving a *transportation problem* [13]. Researchers have incorporated these approaches into partitioners [15], [14], [18].

In [1], Hall shows that the eigenvalue/eignevectors of the Laplacian provide the solution to the one-dimensional weighted quadratic placement problem. The total weighted squared distance between n points can be expressed as:

$$z = \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}(x_i - x_j)^2. \tag{2}$$

Here, x_i is the coordinate assigned to ν_i in a one-dimensional space. The constraint $\Sigma_{i=1}^{n} x_i^2 = 1$ is imposed to avoid the trivial solution in which all x_i's are zeros. If the solution

$$Q = \begin{bmatrix}
3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 3 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & -1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3
\end{bmatrix}.$$

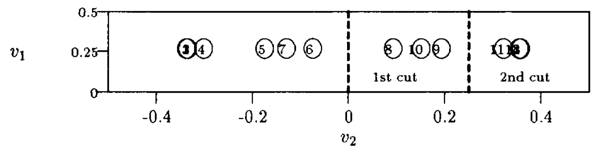


Fig. 2. Ratio-cut bipartition based on linear embedding of the 2nd eigenvector of the Laplacian of the 14-node simple graph. The dashed lines show the cuts at successive bipartitions.

is expressed as a column vector $x = [x_i]$, then the one-dimensional minimization problem (2) can be rewritten in matrix notation (quadratic form) as,

$$\text{minimize } z = x^T Q x \text{ subject to } x^T \cdot x = 1. \quad (3)$$

The eigenvectors of Q provide the solutions of this constrained minimization problem. The value of the solution v_i is the corresponding eigenvalue; $\lambda_i = v_i^T Q v_i$. There are n possible eigenvalues for a system of n linear equations, $\lambda_1 \leq \lambda_2 \leq \cdots \lambda_n$. For a connected graph, the Laplacian has rank of $n-1$, and the minimum eigenvalue $\lambda_1 = 0$ yields the trivial solution $v_1 = ([1/\sqrt{n}], \cdots, [1/\sqrt{n}])$. The 2nd smallest eigenvalue, λ_2, is a lower bound on a nontrivial solution to (3). For example, Fig. 2 shows the one-dimensional placement of the 14-node graph in Fig. 1 according to the 2nd eigenvector of its Laplacian.

In his paper, Hall suggested a k-dimensional generalization in which the eigenvectors are used as the basis for geometric clustering [1].[1] In the next section, we shall re-examine the generalization to k dimensions to motivate the use of the cosine of the angle between the vertices in the k-dimensional embedding, rather than their proximity. This is the basis of the method presented in Section IV. Recently, Hagen and Kahng [20] established a connection between Hall's formulation and 2-way ratio-cut partitioning. Hagen and Kahng construct a 2-way partition from v_2 (the corresponding eigenvector of λ_2) by sorting v_2 and identifying a cut in the sorted v_2 that yields the best ratio-cut cost. This heuristic achieves 17% improvement over the iterative-based ratio-cut 2-way partitioning on the MCNC benchmarks. Fig. 2 shows the 3-way partitioning of the 14-node graph (see Fig. 1) based on the 2nd eigenvector of the Laplacian of the graph.

Any spectral 2-way algorithm can be recursively applied to achieve k-way partitioning, however this requires the reconstruction of the Laplacian and an eigenvalue/eigenvector computation for each round of 2-way partitioning (with the exception of the method of Hendrickson and Leland for up to 8 partitions [21]). Moreover, this can entail a loss of solution quality, as noted in [22]. For example, in the graph shown in Fig. 1, the optimal 2-way partition (1st cut) precludes the optimal 3-way partition. In some sense, the algorithm only uses a fraction of the global information embedded in the eigenvectors. In practice, recursive bisection has been applied to partition unstructured meshes for parallel computation [23] and static load balancing of scientific computations [21].

[1] As pointed out by Professor Andrew B. Kahng, Hall's method is closely related to what is known as Karhunen and Loèeve (KL) transform in cluster or factor analysis for extracting principal components [19]. Also, the Karhunen and Loèeve (KL) transform can find its origin due to the work of Pearson in "On lines and planes of closest fit to systems of points in space," *Philo. Mag.*, 1901.

III. Spectral k-Way Ratio-Cut Partitioning

This section presents a *new* spectral k-way ratio-cut partitioning formulation that generalizes Simon, Pothen, and Liou [18] and Hagen and Kahng's [20] spectral 2-way partitioning to k-way. We begin by generalizing Hall's one-dimensional weighted quadratic placement optimization to k dimensions and then relate the placement optimization problem to spectral k-way partitioning. A new ratio-cut cost metric is introduced, and its relation to the generalized Hall's weighted quadratic placement formulation is established to provide a lower bound on the cost metric.

Our formulation is an advancement over the spectral 2-way partitioning [18], [20] for several reasons. First, all k partitions are found simultaneously; it is therefore using all the global information available in the eigenvectors. Second, the orthonormality among the eigenvectors forms the basis of our heuristics to coerce the placement into partitions. This method is therefore taking the full advantage of the symmetric eigenvalue solver: the Lanczos algorithm [24].

A. K-Way Weighted Quadratic Placement

Let X be an $n \times k (1 \leq k \leq n)$ matrix $X = [x_{ij}] = [x_1, \cdots, x_k]$. The sum of the weighted quadratic placement cost, z, is

$$z = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{h=1}^{k} a_{ij} (x_{ih} - x_{jh})^2. \quad (4)$$

With a little bit of algebra, it can be shown that $z = trace(X^T Q X)$. (The $trace(M)$ of a square matrix M is the sum of its diagonal entries.) This establishes a connection between the weighted quadratic placement and the quadratic form $X^T Q(G) X$. The next theorem suggests an embedding X to minimize z.

Theorem 1: The $n \times k$ matrix that minimizes the $trace(X^T Q(G) X)$ subject to the constraint $X^T X = I_k$ is the $n \times k$ matrix whose k columns consist of the k eigenvectors of Q corresponding to the k smallest eigenvalues of Q.

If $V = [v_1, \cdots, v_k]$ is formed with any k eigenvectors of Q, then $QV = V\Lambda_k$ where Λ_k is the diagonal matrix formed with the k eigenvalues corresponding to the k eigenvectors in V. This means that

$$V^T Q V = V^T V \Lambda_k = I_k \Lambda_k = \Lambda_k,$$

and $trace(V^T Q V) = trace(\Lambda_k)$, which is the sum of the eigenvalues. To minimize $trace(V^T Q V)$, the k eigenvectors corresponding to the k smallest eigenvalues should be selected. It remains to show that there exists no matrix X satisfying $X^T X = I_k$ with $trace(X^T Q X) < trace(\Lambda_k)$, where Λ_k is formed with the k smallest eigenvalues. Suppose there is such an X. Let k be the smallest integer for which such an X exists. Since Q is positive semi-definite, i.e., $x^TQx \geq 0 = \lambda_1$ for all x, k must be greater than 1. Let y be the vector with $||y|| = 1$ in the subspace spanned by X, which maximizes y^TQy. We can find an orthonormal basis for this subspace, B, which has y as one of its vectors. Since X and B span the same subspace and $B^TB = I_k$, there exists a $k \times k$ matrix S satisfying $B = XS$ and $S^{-1} = S^T$. Since similar matrices have the same eigenvalues and $B^TQB = S^{-1}(X^TQX)S$ and X^TQX are similar matrices, we have

$$trace(B^TQB) = trace(X^TQX) < trace(\Lambda_k).$$

Let B_* be the $n \times (k-1)$ matrix obtained by removing the column of B corresponding to y. Then, we must have $trace(B_*^TQB_*) \geq trace(\Lambda_{k-1})$, since k was the smallest integer for which this inequality did not hold. Then

$$\begin{aligned} y^TQy &= trace(B^TQB) - trace(B_*^TQB_*) \\ &< trace(\Lambda_k) - trace(\Lambda_{k-1}) = \lambda_k. \end{aligned}$$

But this implies that

$$\max_{\substack{x \in X \\ ||x||=1}} x^TQx = \max_{\substack{x \in X \\ x \neq 0}} \frac{x^TQx}{x^Tx} = y^TQy < \lambda_k,$$

which violates the Courant-Fisher Theorem [25, p. 101], which states that the k^{th} smallest eigenvalue,

$$\lambda_k = \min_{dim(E)=k} \max_{0 \neq x \in E} \frac{x^TQx}{x^Tx}.$$

This implies that all matrices X with $X^TX = I_k$ have $trace(X^TQX) \geq trace(\Lambda_k)$. □

Theorem 1 is a special case of Fan's Theorem [26] when Q is a real symmetric matrix. Another short proof of Fan's Theorem is in [27]. Hence, V, the smallest k eigenvectors of Q, provide the solution to

$$\sum_{i=1}^{k} \lambda_i = trace(V^TQV) = \min_{X^T \cdot X = I_k} trace(X^TQX). \quad (5)$$

As an example, Fig. 3 shows a 2-D embedding based on the 2[nd] and 3[rd] eigenvectors of the Laplacian of the graph depicted in Fig. 1. Recall that v_1 is the constant $1/\sqrt{n}$ for all nodes.

B. Relation to Spectral K-Way Partitioning

A k-way partition of the nodes of graph $G, \mathcal{P} = \{P_1, P_2, \cdots P_k\}$ can be represented by an $n \times k$ *ratioed assignment matrix* $R = [r_{ih}]$ where

$$r_{ih} = \begin{cases} \dfrac{1}{\sqrt{|P_h|}} & \text{if } \nu_i \in P_h \\ 0 & \text{if } \nu_i \notin Ph \end{cases} \quad (6)$$

The rows no longer necessarily sum to 1, and column h now sums to $\sqrt{|P_h|}$. The related $n \times n$ ratioed partitioned matrix $P_R = [rp_{ij}]$ is defined by

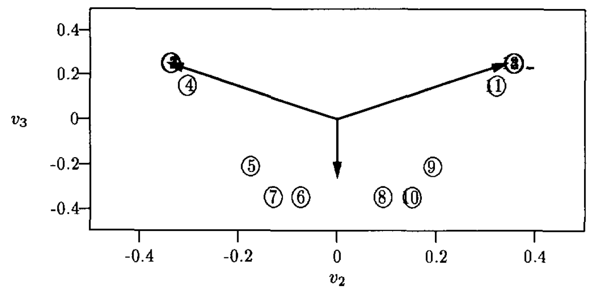


Fig. 3. Two-dimensional embedding based on the 2[nd] and 3[rd] eigenvectors of the Laplacian of the 14-node graph as shown in Fig. 1. The arrows indicate the "prototypes" as discussed in Section IV-A.

$$rp_{ij} = \begin{cases} \dfrac{1}{|P_g|} & \text{if } \nu_i \text{ and } \nu_j \text{ both belong to } P_g \\ 0 & \text{otherwise.} \end{cases}$$

The following properties of R and P_R are needed both to derive the lower bound on the k-way ratio-cut cost metric and to extract the partitions from the k eigenvectors. They follow directly from the definitions of R and P_R. Let E_h be the total weight of the edges in G having exactly one endpoint in P_h.

Lemma 1: If R is the ratioed assignment matrix for $\mathcal{P} = \{P_1, P_2, \cdots, P_k\}$ then

1) $trace(R^TQR) = \Sigma_{h=1}^{k} (E_h/|P_h|)$.
2) $RR^T = P_R$, *the ratioed partition matrix* .
3) $R^TR = I_k$.

We are now in the position to establish a connection between spectral k-way ratio-cut partitioning and (Hall's) generalized weighted quadratic placement formulation.

Theorem 2: The sum of the smallest k eigenvalues of the Laplacian Q of a weighted graph G is a lower bound on $\Sigma_{h=1}^{k} (E_h/|P_h|)$ for any k-way partition of G.

Proof: Let Π be the set of all k-way partitions of a graph G and let $R_{\mathcal{P}}$ denote the the ratioed assignment matrix associated with a particular $\mathcal{P} \in \Pi$. Since by Lemma 1 we know that $R_{\mathcal{P}}$ satisfies the constraint on X in (5),

$$\begin{aligned} \sum_{i=1}^{k} \lambda_i &= \min_{X^TX=I} trace(X^TQX) \\ &\leq \min_{\mathcal{P} \in \Pi} trace(R_{\mathcal{P}}^TQR_{\mathcal{P}}) = \min_{\mathcal{P} \in \Pi} \sum_{h=1}^{k} \frac{E_h}{|P_h|}. \quad (7) \end{aligned}$$

□

C. A New K-Way Ratio Cut Cost Metric

We define a generalized k-way ratio-cut cost metric of a k-way partition $\mathcal{P}$ of graph G to be

$$cost(\mathcal{P}) = \sum_{h=1}^{k} \frac{E_h}{|P_h|}. \quad (8)$$

For $k = 2, E_c = E_1 = E_2$ so we have $cost(\mathcal{P}) = E_1/|P_1| + E_2/|P_2| = n(E_c/|P_1| \cdot |P_2|)$. Thus, $cost(\mathcal{P})$ is a generalization from 2 to k partitions of the ratio-cut cost metric defined by [2] after scaling by n. For $k = 2$, since $\lambda_1 + \lambda_2 = 0 + \lambda_2 = \lambda_2$, from (7), we obtain $\lambda_2 \le cost(\mathcal{P}) = E_c/|P_1| + E_c/|P_2| = n(E_c/|P_1|\cdot|P_2|)$, which is the lower bound derived by [18] and [20].[2] To conform to the classical ratio-cut metric we define a scaled version of (8)

$$scost(\mathcal{P}) = \frac{1}{n(k-1)} cost(\mathcal{P}). \quad (9)$$

The analysis in [2] suggested the merit of having a cost function that is constant for random graphs. For a random graph with edge probability f, and any k-way partition $\mathcal{P}$, our scaled cost metric yields $scost(\mathcal{P}) = f$. Recently Yeh, Cheng, and Lin proposed a different generalization of the ratio-cut cost metric from 2 to k partitions [28]:

$$\frac{\frac{1}{2}\sum_{h=1}^{k} E_h}{\sum_{h=1}^{k-1}\sum_{g=h+1}^{k} |P_h| \cdot |P_g|}. \quad (10)$$

Both generalizations (9) and (10) are constant for complete graphs with random edge weights. Both reward partitions with reduced cut weights and balanced partition sizes. However, they differ according to the distribution of the cut edges among the partitions. Cost metric (9) favors partitions in which the smaller-sized partitions have smaller weight cuts, while cost metric (10) only varies according to the total weight of the edges cut, regardless of their distribution.

IV. From Embeddings to Partitions

Theorem 2 justifies the approximation of R by the first k eigenvectors $V = [v_1, \cdots v_k]$. Our approach to k-way partitioning is to "reverse engineer" the partitions from V. If V is an approximation to R, then an approximation to P_R is $Z = VV^T$ since by Lemma 1 $P_R = RR^T$. Each z_{ij} element in Z is the dot product of the i^{th} and j^{th} rows of V. If $V^T = [v'_1, \cdots, v'_n]$, then $z_{ij} = v_i'^T \cdot v'_j$. When ν_i and ν_j are in the same partition, P_h, the ij^{th} entry of P_R is $1/|P_h|$ while it is 0 when they are in different partitions. In contrast the ij^{th} entry of P is either 1 or 0. For this reason recognizing the partitions from an approximation to P rather than P_R is easier. To this end we exploit the following relation between P and P_R. Given an $n \times n$ matrix M, let $N(M)$ be the $n \times n$ diagonal matrix where n_{ii} is the reciprocal of the norm of row i of M. Then we have $P = N(R) \cdot RR^T \cdot N(R) = N(P_R) \cdot P_R$. Performing the same normalization on Z gives us: $\hat{P} = N(V) \cdot VV^T \cdot N(V)$. In this case $N(V)$ is a diagonal matrix with

$$n_{ii} = \frac{1}{\sqrt{\sum_{h=1}^{k} v_{ih}'^2}}, \quad \text{and} \quad \hat{P}_{ij} = \frac{v_i'^T \cdot v'_j}{||v'_i|| \cdot ||v'_j||},$$

which is the cosine of the angle between the two row vectors i and j of V (or the column vectors of V^T). These *directional cosines* provide a measure of how close the vertices are from each other.

[2]Also referred to by Mohar as a folk theorem [12].

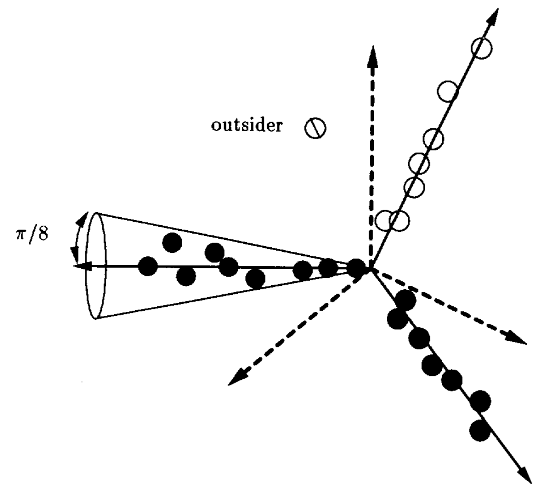


Fig. 4. Spectral K-Way (KP) clustering heuristic, representation of an outsider, a framework, and base vectors.

To summarize our method, we "approximate" the ratioed assignment matrix R by the eigenvectors V. The partition matrix P is recovered from a projector (VV^T) formed by the eigenvectors after normalization:

$$(V \approx R) \Longrightarrow (\hat{P} \approx P)$$

$$Q \overset{Eigensolver}{\Longrightarrow} V \overset{\substack{directional\\cosine}}{\Longrightarrow} \hat{P} \overset{clustering}{\Longrightarrow} P.$$

Since it is expensive to compute all n^2 entries of $\hat{P}$ for graphs with $\approx 10^3$ nodes, a heuristic will be presented in the next section that computes the directional cosines on an as-needed basis; only a small fraction of the elements in $\hat{P}$ are actually computed.

A. A Time and Space Efficient Clustering Heuristic: KP

In this section, we present a clustering heuristic to form partitions without calculating the entire $\hat{P}$ matrix. The cosine of the angle between rows i and j of the V matrix $\cos(v'_i, v'_j)$ are computed only when needed. The goal is to "reverse engineer" the partitions under the premise that $\cos(v'_i, v'_j)$ is close to 1 when vertices ν_i and ν_j are in the same partition and 0 otherwise.

The first step is to select k vertices to serve as the *prototypes* (seeds) of the k partitions. In Fig. 4 the prototypes are depicted by solid arrows. In the first iteration, the k prototypes are selected by magnitude and (near) orthogonality to all previously selected vertices. In subsequent four clustering iterations, the average (in terms of coordinate location) of each of the k partitions from the previous iteration is computed and the closest vertex to the average is selected as the new seed.

After selecting the prototypes, all vertices whose directional cosine is within $\cos(\pi/8)$ of a prototype are added to that prototype's partition. The remaining vertices (*outsiders*) are considered one at a time and merged into the partitions. When the graph does not have a "natural" k-way partition, the cost of the result depends largely on how the outsiders are merged into existing partitions. The number of hyperedge cuts from each outsider to each existing partition is calculated, and the partition with the largest number of cuts is chosen as the target partition for that vertex. Using a heap, the outsider with the largest key is merged into its target partition and its still-unassigned neighbors have their hyperedge cuts recalculated and their keys adjusted. This is repeated until all outsiders have been assigned to a partition.

One problem with this method is that the outsiders may share more edges among themselves than with any of the existing partitions. To allow for this possibility, after each assignment of an outsider, we also consider merging *en masse* the entire set of remaining outsiders with each of the k partitions. The costs of these *en masse* mergings may be efficiently computed incrementally in $O(bk)$ time, where b is the maximum degree of a vertex. The result with the best ratio-cut cost is recorded, and then the algorithm proceeds to consider the next outsider at the top of the heap. The time required is $O(n(bk^2 + bk\log(n)))$ per iteration and this clustering algorithm uses $O(1)$ *additional* space per vertex, so summing over all vertices, we need only $O(n)$ additional space. The total space is dominated by the size of the eigenvector V matrix, which is $O(nk)$ space.

V. Experimental Results

A. K-Way Partitioning Versus Bipartitioning

We compare our results with successive spectral 2-way partitioning. We reimplemented Hagen and Kahng's spectral 2-way partitioning algorithm with the EIG1 heuristic (which evaluates and retains the best partitioning of the vertices at *every* point in their ordering by the 2nd eigenvector) [20]. We refer to this algorithm as **SB** and to our k-way spectral ratio-cut partitioning algorithm as **KP**. We used 4 iterations of the clustering heuristic.

We ran both algorithms on the graphs derived from the MCNC layout synthesis benchmarks. The hypergraphs of the benchmarks were preprocessed to remove high-fanout nets of degree greater than 99 and then clique-expanded into graphs. Nets whose degree were greater than 99 were removed in order to reduce storage space and processing time, since a clique expansion is performed to transform hypergraphs into graphs. However, these high-fanout nets are considered in the final ratio-cut metric. Specifically, a net of degree d (a hyperedge with d vertices) is modeled by a clique of $d(d-1)/2$ edges with weights. Researchers have chosen $w = 1/d$ or $w = 1/(d-1)$ with the rationale that the total number of connections needed in the worst case is $d-1$ and d, respectively. Frankle proposed using

$$w = \left(\frac{2}{d}\right)^{3/2}$$

for the purpose of minimizing the worst-case deviation of cost from span^2 [16].

TABLE I
Summary of Percentage of Improvement of KP Versus SB Results Using Different Net Models

Net Models	$\frac{1}{Degree-1}$	$\frac{1}{Degree}$	$(\frac{2}{Degree})^{\frac{3}{2}}$
Graph	% Imprv	% Imprv	% Imprv
p1	40.06	36.10	40.00
p2	-0.41	-0.03	-13.76
t2	0.35	15.87	14.90
t3	10.84	10.90	1.29
t4	30.56	28.75	32.35
t5	-6.89	-3.68	6.55
t6	7.99	-2.00	8.98
Average	11.78	12.27	12.90

TABLE II
Detailed Results With Frankle's Net Model

Graph	*k*	*Ratio-cut cost* $\times 10^{-5}$			*time* (sec on DEC 5000/240)	
		SB	KP	% Imprv.	SB	KP
p1	2	13.525	13.525	0.000	32.63	112.0+1.15
	5	40.276	29.941	34.516	113.62	112.0+3.04
	25	86.167	75.153	14.656	367.61	112.0+16.99
p2	2	4.776	4.644	2.835	154.44	381.8+12.68
	5	8.469	10.061	-15.824	475.53	381.8+21.07
	25	16.485	23.065	-28.528	1292.94	381.8+78.90
t2	2	8.730	9.256	-5.689	90.43	257.2+3.20
	5	14.463	13.865	4.311	318.95	257.2+14.60
	25	42.043	29.094	44.505	791.16	257.2+45.49
t3	2	31.153	31.153	0.000	68.17	243.1+2.54
	5	16.921	17.482	-3.212	253.28	243.1+6.68
	25	32.797	28.722	14.190	747.92	243.1+34.39
t4	2	66.050	66.050	0.000	82.04	280.8+2.66
	5	28.189	14.289	97.277	280.29	280.8+8.00
	25	35.953	31.618	13.709	685.81	280.8+32.69
t5	2	8.938	11.344	-21.216	150.26	594.8+5.01
	5	6.281	7.977	-21.265	598.43	594.8+15.43
	25	16.942	16.887	0.325	1562.29	594.8+69.16
t6	2	28.571	28.571	0.000	175.17	299.4+6.19
	5	15.105	15.011	0.627	694.79	299.4+13.35
	25	35.880	38.565	-6.962	1519.25	299.4+94.24

The KP and SB programs were tested on the MCNC benchmarks (p1, p2_ga, t2, t3, t4, t5) using k ranging from 2 to 25. Due to space limitation, only a summary of the experimental results are given in Tables I and II; please refer to [29] for details. Overall, KP improves the ratio-cut metric by 12%. On run time, KP starts up slow (e.g. 112 sec for p1) because it computes all 25 eigenvalues/eigenvector pairs first, but subsequent clusterings of the eigenvectors for different k's reuse the eigenvectors. In contrast, successive bipartitioning (SB) requires an eigenvector computation for each value of k. For large k, KP is much faster than SB.

The effectiveness of both partitioning algorithms can be demonstrated visually. Figs. 5 and 6 plot the embedding of the vertices implied by eigenvectors v_2' and v_3' of the graph of p1. Each dot in the diagram corresponds to a vertex in the graph. This wishbone-like plot in Fig. 5 has three distinctive components. The three partitions are identified by

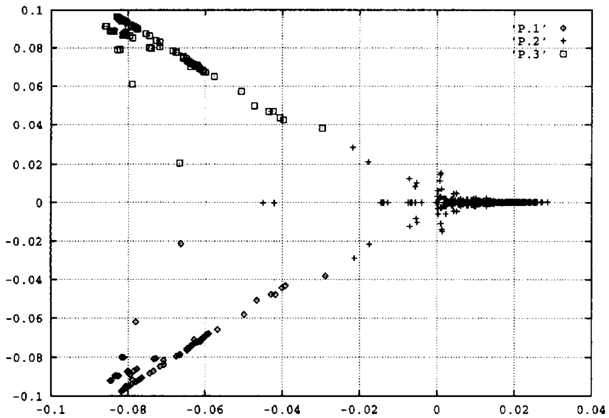


Fig. 5. Embedding with $v_2'v_3'$ for benchmark p1 with "◇," "+," and "□" indicating the 3 partitions obtained by KP.

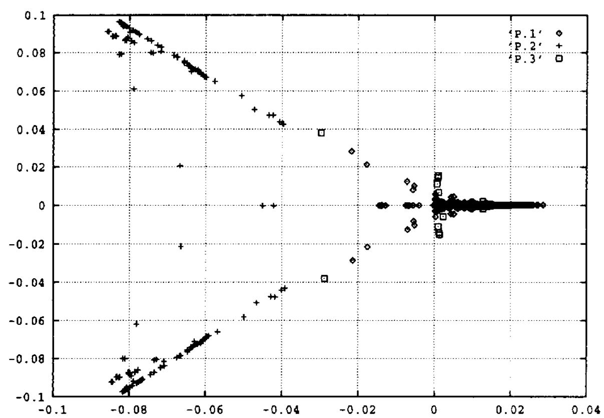


Fig. 6. Embedding with $v_2'v_3'$ for benchmark p1 obtained by SB.

our KP clustering algorithm and are marked as "◇," "+," and "□," respectively. Fig. 6 shows the 3-way partitioning of the same graph by SB. For this benchmark, it is evident that SB produces inferior partitions to those produced by KP due to a lack of global perspective. SB dissected the "wishbone" into three pieces. However, SB does produce better results on graphs that are less *structured* (e.g. t5), because SB's heuristic can afford to evaluate many possible partitions.

B. Spectral Analysis

This raises the question of when to apply k-way partitioning or successive bipartitioning. The spectrum $\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ of a graph can be a quantitative measure of the structure of a graph. Consider the lower bound of a k-way partition $\mathcal{P}$ derived earlier:

$$scost(\mathcal{P}) \geq \frac{\sum_{h=1}^{k} \lambda_h}{n(k-1)}.$$

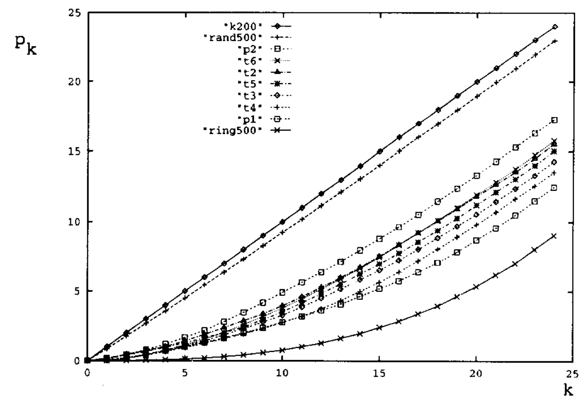


Fig. 7. Spectra of different graphs, ρ_k versus k.

We can use this lower bound as a measure of "structure." We use ρ_k as an index which measures the unstructuredness of a graph for a k-way partitioning. Let

$$\rho_k = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_{25}}.$$

For our tests, we normalized the eigenvalues of each graph with its 25^{th} eigenvalue. Graphs that have small ρ_k are well-structured, while graphs that have large ρ_k are not well-structured. Fig. 7 plots the ρ_k-curves for all of the MCNC graphs using Frankle edge weights. We can validate the claim that ρ_k is a reasonable measure of structure by considering several special graphs:

- Complete graph k200 has 200 vertices. Its ρ_k-curve is the highest since it has no structure.
- Random graph rand500 has 500 vertices, each edge of the complete graph had a 10% chance of being included when the graph was generated.
- A ring is one of the simplest regular structures, and it has the lowest ρ_k-curve among the graphs we tested.

The two graphs in which KP surpassed SB by the largest margin were p1 and t4. These two graphs have the lowest two curves in Fig. 7. The graphs t2, t3, and t5 were the intermediate curves, and the win/loss margin of KP over SB was small. Finally, the curve for graph p2 was the highest of the MCNC tests, and it was also the graph in which KP performed the worst.

C. Partition Size Distribution

Another important issue is the size of the partitions generated by a partitioning problem. By using a k-way ratio-cut cost metric, partition sizes are balanced with the number of nets cut automatically. Ideally, one would like all partitions to be near the median-size range. Figs. 8 and 9 show the distribution of partition sizes. The horizontal axis is the partition size, while the vertical axis gives the number of partition instances that were of that size. The statistics were found by extracting all of the partitions for all of the test files for $2 \leq k \leq 25$ from

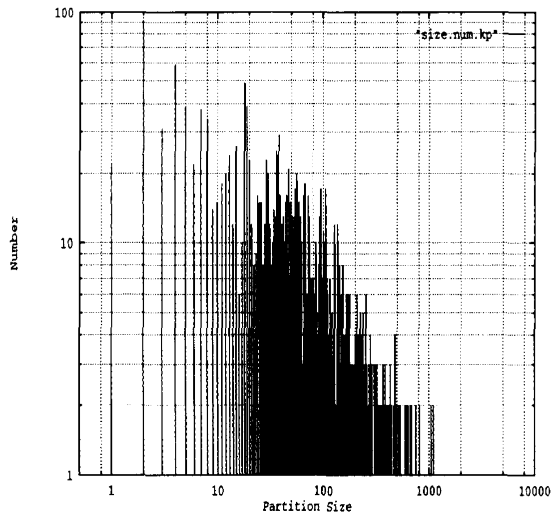


Fig. 8. Partition size distribution for KP (Spectral K-Way Partitioning).

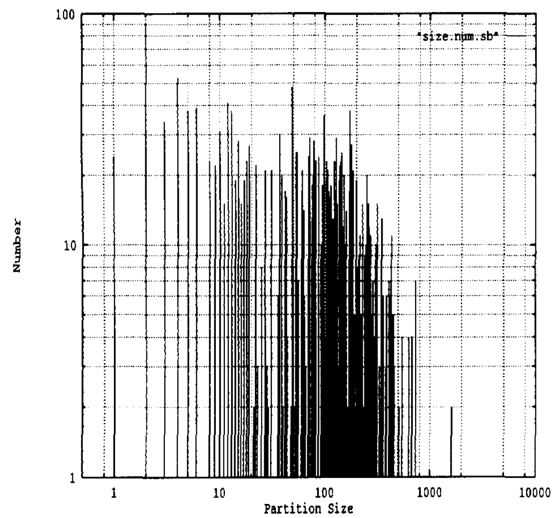


Fig. 9. Partition size distribution for SB (Successive Bipartitioning).

the graphs using Frankle edge weights. We find that there are actually a substantial number of partitions of extremely small (<20 nodes) size generated by both SB and KP. This brings out one of the weaknesses of the k-way ratio-cut metric—the tendency toward producing a number of small partitions.

VI. Conclusion

We have developed a general theory of spectral ratio-cut k-way partitioning. Our theory provides the backbone for efficient heuristics to find k-way partitions of a graph based on the eigenvectors of the Laplacian of the graph. We have presented some experimental results to support our development of the theory and heuristics. Our method compares favorably to the results obtained by recursive spectral 2-way partitioning.

Alpert and Kahng recently applied geometrical embedding technique to cluster the eigenvectors to form k-way partitions [30]. They evaluated their partitioning result using our new ratio-cut cost metric. Judging from their result published in [30, Table III], it doesn't seem to be an improvement over our method.

Acknowledgment

The authors have benefited from discussions with Professor Andrew B. Kahng of UCLA and Dr. Horst D. Simon of NASA Ames Research Center.

References

[1] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Sci.*, vol. 17, pp. 219–229, Nov. 1970.
[2] C.-K. Cheng and Y.-C. A. Wei, "An improved two-way partitioning algorithm with stable performance," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1502–1511, Dec. 1991.
[3] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Tech. J.*, vol. 49, 1970.
[4] C. M. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. ACM IEEE 19^{th} Design Automation Conf.*, June 1982, pp. 175–181.
[5] L. A. Sanchis, "Multiple-Way Network Partitioning," *IEEE Trans. Comput.*, vol. C-38, pp. 62–81, Jan. 1989.
[6] B. Krishnamurthy, "An Improved Min-Cut Algorithm For Partitioning VLSI Networks," *IEEE Trans. Comput.*, vol. C-33, pp. 438–446, May 1984.
[7] T. Leighton and S. Rao, "An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximate algorithms," *FOCS 29*, 1988.
[8] C. K. Cheng and T. C. Hu, "The optimal partitioning of networks," Tech. Rep. CS89-146, University of California, San Diego, Mar. 1989.
[9] R. Boppana, "Eigenvalues and graph bisection: An average-case analysis," in *Proc. IEEE Symp. on Foundations of Computer Science*, 1987, pp. 280–285.
[10] D. M. Cvetkovic, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Application.* New York: Academic Press, Inc., 1979.
[11] M. Fiedler, *Special matrices and their applications in numerical mathematics.* Martinus Nijhoff Publishers, 1986.
[12] B. Mohar and S. Poljak, *Eigenvalues in Combinatorial Optimization.* University of Ljubljana, Ljubljana, Slovenia, 1993.
[13] E. R. Barnes, "An algorithm for partitioning the nodes of a graph," *SIAM J. Algorithm and Discrete Method,* vol. 3, pp. 541–550, Dec. 1982.
[14] W. Donath and A. Hoffman, "Lower bounds for the partitioning of graphs," *IBM J. Research and Development,* pp. 420–425, 1973.
[15] S. W. Hadley, B. L. Mark, and A. Vanelli, "An Efficient Eigenvector Approach for Finding Netlist Partitions," *IEEE Trans. Computer-Aided Design*, vol. CAD-11, pp. 885–892, July 1992.
[16] J. Frankle and R. M. Karp, "Circuit placements and cost bounds by eigenvector decomposition," in *IEEE Int. Conf. Computer-Aided Design ICCAD-86*, Nov. 1986, Santa Clara, CA, pp. 414–417.
[17] F. Rendl and H. Wolkowicz, "A projection technique for partitioning the nodes of a graph," Tech. Rep., University of Waterloo, Apr. 1991.
[18] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM J. Matrix Analysis and Appl.*, vol. 11, no. 3, pp. 430–452, 1990.
[19] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentice-Hall Advanced Reference Series, 1988.
[20] L. Hagen and A. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Computer-Aided Design,* vol. CAD-11, pp. 1074–1085, Sept. 1992.
[21] B. Hendrickson and R. Leland, *Multidimensional Spectral Load Balancing.* Sandia National Laboratories, Albuquerque, NM 87185, Jan. 1993.
[22] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout.* Wiley-Teubner, 1990.
[23] S. T. Barnard and H. D. Simon, *A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems.* NASA Ames Research Center, Apr. 1993.
[24] B. N. Parlett and D. S. Scott, "The Lanczos algorithm with selective orthogonalization," *Mathematics and Computations*, vol. 33, no. 11, pp. 217–238, 1979.
[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford Science Publications, 1965.

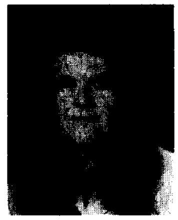
[26] K. Fan, "On a theorem of Weyl concerning eigenvalues of linear transformation," *Proc. National Academy of Sciences, USA,* vol. 35, pp. 652–655, 1949.

[27] M. L. Overton and R. S. Womersley, "On the sum of the largest eigenvalues on a symmetric matrix," *SIAM J. Matrix Analysis and Appl.*, vol. 13, no. 1, pp. 41–45, 1992.

[28] C.-W. Yeh, C.-K. Cheng, and T.-T. Y. Lin, "A probabilistic multicommodity-flow solution to circuit clustering problems," in *Proc. IEEE Int. Conf. Computer-Aided Design ICCAD-92,* Nov. 1992, Santa Clara, CA.

[29] J. Y. Zien, "Spectral k-way ratio cut partitioning," Master's Thesis, University of California, Santa Cruz, Mar. 1993.

[30] C. J. Alpert and A. B. Kahng, "Geometric embeddings for faster and better multi-way netlist partitioning," in *Proc. ACM IEEE 30^{th} Design Automation Conf.*, June 1993, Dallas, TX, pp. 743–748.



Martine D. F. Schlag (S'85–M'86) received the B.A. degree in mathematics and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles in 1978, 1982, and 1986, respectively. She is currently associate professor in Computer Engineering at the University of California, Santa Cruz. She held summer positions at the IBM T. J. Watson Research Center in 1982 and 1983 and was Assistant Professor in Computer Science at the University of Washington from 1986 to 1988. In 1987, she received an NSF Presidential Young Investigator Award. Her interests are in theoretical computer science, computational geometry, VLSI design tools, VLSI complexity, formal specification language, and Field-Programmable Gate Arrays. She is serving as Associate Editor for the *IEEE Transactions on Computers.*

Pak K. Chan (S'77–M'83) received the B.Sc. degree in Electronics from the Chinese University of Hong Kong and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles in 1984 and 1987.

During the summer of 1986, he worked at the IBM T. J. Watson Research Center. Presently, he is an associate professor of Computer Engineering at the University of California, Santa Cruz. His research interests include timing analysis, circuit theory, computer arithmetic, algorithms, and Field-Programmable Gate Arrays. He is serving as Associate Editor for the *IEEE Transactions on Computers.*

Jason Y. Zien received his B.S. in computer engineering from Carnegie Mellon University, Pittsburgh, PA in 1991, and his M.S. in computer engineering from the University of California, Santa Cruz, in 1993. He was a summer intern at Xilinx in 1993. His current research interests include partitioning, clustering, computer graphics, and scientific computing.