Multi-level Spectral Hypergraph Partitioning with Arbitrary Vertex Sizes

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

Computer Engineering
University of California at Santa Cruz
Santa Cruz, CA 95064

Abstract

This paper presents a new spectral partitioning formulation which directly incorporates vertex size information. The new formulation results in a generalized eigenvalue problem, and this problem is reduced to the standard eigenvalue problem. Experimental results show that incorporating vertex sizes into the eigenvalue calculation produces results that are 50% better than the standard formulation in terms of scaled ratio-cut cost, even when a Kernighan-Lin style iterative improvement algorithm taking into account vertex sizes is applied as a post-processing step. To evaluate the new method for use in multi-level partitioning, we combine the partitioner with a multi-level bottom-up clustering algorithm and an iterative improvement algorithm for partition refinement. Experimental results show that our new spectral algorithm is more effective than the standard spectral formulation and other partitioners in the multi-level partitioning of hypergraphs.

1 Introduction

Previous spectral algorithms for partitioning graphs and hypergraphs have been limited by the fact that they implicitly assume that all vertices in a graph are the same size. In some problems, such as the partitioning of logic blocks for field-programmable gate arrays, this assumption may be valid, however, in problems such as macro cell partitioning or partitioning with hierarchical clustering, vertices are unlikely to all have the same size.

In this paper, we present a new spectral partitioning formulation which directly incorporates vertex sizes. This formulation yields a generalized eigenvalue problem that can be reduced to the standard eigenvalue problem. Thus, existing standard eigenvalue computation code can be used with no modifications. We present benchmarks to quantitatively show the effectiveness of the new method.

Multiple levels of bottom-up clustering reduce the problem size and tend to produce superior results. Multi-level algorithms have been developed using both spectral [2], and iterative [9, 10] approaches. However after the application of hierarchical clustering algorithms, the resulting graph may contain vertices of different sizes. An effective spectral partitioner must take those sizes into account to produce good solutions. Our spectral partitioner, MP (Multi-level K-way Partitioner), does exactly that.

Spectral algorithms were first proposed for placement and partitioning by Hall [8]. Fast bipartitioning methods were developed based on a linear ordering of the vertices using the eigenvector associated with the second smallest eigenvalue of the Laplacian of a graph in [11, 7]. A k-way spectral partitioning algorithm and new k-way ratio-cut cost function was presented in [4]. Other methods for spectral k-way partitioning are presented in [1, 3].

A multi-level partitioning algorithm using recursive applications of the ratio-cut metric to form clusters is presented in [12]. The first spectral multi-level algorithm was developed by [2], however, it did not include iterative refinement at successive levels. Hendrickson and Leland implemented a multi-level spectral algorithm with a k-way Kernighan-Lin style refinement algorithm [9]. An in-depth study [10] of various multi-level contraction, initial partitioning, and refinement strategies on meshes concludes that all of the methods tested perform nearly equally as well.

2 Spectral partitioning with vertex sizes

We describe a new spectral method in this section. We present the definitions and then formulate the problem.

Given a graph with a set of n vertices, V, we wish to find k partitions of this graph. A *partitioning* of the graph is a division of the n vertices into k disjoint, non-empty subsets $P_1, P_2, \cdots, P_k$ such that $V = P_1 \cup P_2 \cup \cdots \cup P_k$. The $n \times n$ *adjacency matrix* of the graph, A, has entries a_{ij} which are the sum of the weights of the edges between vertices i and j. The $n \times n$ diagonal *degree* matrix, D, has entries d_{ii} equal to the sum of the weights of all edges on vertex i. The *Laplacian* matrix is defined as $Q = D - A$. E_h is the sum of the weights of the edges which have exactly one vertex in partition h. $||P_h||$ is the sum of the sizes of all vertices in partition h. A partitioning can be represented by

*Supported in part by National Science Foundation Grant MIP-9223740

ICCAD '96
1063-6757/96 $5.00 © 1996 IEEE

an $n \times k$ *assignment* matrix Y where y_{ih} is 1 when vertex i is in partition h and y_{ih} is 0 otherwise. Given a partitioning, the $n \times k$ *ratioed assignment* matrix, R, has as entry r_{ih} the value $\frac{y_{ih}}{\sqrt{||P_h||}}$. This definition differs slightly from the one in [4]. M is the $n \times n$ diagonal matrix whose m_{ii} entry is the size of vertex i.

Among the many variations of the k-way partitioning problem, we focus on optimizing the k-way ratio-cut cost function [4], that is, finding a solution R such that $\sum_{h=1}^{k} \frac{E_h}{||P_h||}$ is minimized. Although it may appear to be the same cost function as presented in [4], there is a subtle difference: the definitions of R and $||P_h||$ include the actual vertex sizes rather than the number of vertices in a partition.

As in the proof of Lemma 1 found in [4]: we can show that the h^{th} diagonal entry of R^TQR satisfies:

$$(R^TQR)_{hh} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}(r_{ih} - r_{jh})^2 = \frac{E_h}{||P_h||} \quad (1)$$

Hence the trace of R^TQR is $\sum_{h=1}^{k} \frac{E_h}{||P_h||}$.

Vertex sizes are implicitly incorporated into our problem by our new definitions of R and $||P_h||$. We can show that by taking vertex sizes into account, the constraint $R^TR = I$ is replaced by $R^TMR = I$. Let $\mathcal{D}_{n,k}$ be the set of $n \times k$ matrices which have a single non-zero entry in every row and for each column, exactly one non-zero value among its non-zero entries. It is derived in [13] that:

> R is a ratioed partition matrix if and only if $R \in \mathcal{D}_{n,k}$ and $R^TMR = I$.

3 Relaxed Problem Formulation

Our objective is to find the matrix which minimizes R^TQR subject to the constraint $R^TMR = I$ and $R \in \mathcal{D}_{n,k}$. This problem is equivalent to the ratio-cut partitioning problem, and hence, there is no known optimal polynomial-time solution. We can, however, relax the problem by removing the $R \in \mathcal{D}_{n,k}$ constraint. The relaxed problem turns out to be a quadratic placement problem [8], which can be solved in polynomial time. The relaxed problem is defined as:

$$\text{minimize } trace(X^TQX) \text{ subject to } X^TMX = I. \quad (2)$$

In the spectral partitioning literature, it is typical to use the constraint: $X^TX = I$. The new constraint, $X^TMX = I$, is intended to utilize any vertex size information that is available. This constraint reduces to $X^TX = I$ when all of the vertices are unit size. Because M is a diagonal matrix, we can transform Eq. (2) into a standard eigenvalue problem. Let $M = S^TS$.[1] Assume that all vertices have positive size. By substituting $\hat{Q} = S^{-1^T}QS^{-1}$ and $\hat{X} = SX$, we transform Eq. (2) to a modified problem,

$$\text{minimize } trace(\hat{X}^T\hat{Q}\hat{X}) \text{ subject to } \hat{X}^T\hat{X} = I. \quad (3)$$

Equation (3) has the same form as previous spectral partitioning formulations, except that $\hat{Q}$ is no longer the Laplacian of the graph. By using the method of Lagrange multipliers or Fan's Theorem as shown in previous literature [8, 4], Eq. (3) leads to the standard eigenvalue problem.

$$\hat{Q}\hat{X} = \hat{X}\Lambda. \quad (4)$$

Theorem 1 Fan's Theorem *[5]*
Let the eigenvalues λ_i of a symmetric matrix Q be so arranged that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. For any positive integer $k \leq n$, the sums $\sum_{i=1}^{k}\lambda_i$ and $\sum_{i=1}^{k}\lambda_{n+1-i}$ are respectively the minimum and maximum of $\sum_{j=1}^{k} x_j^TQx_j$ when k orthonormal vectors x_j $(1 \leq j \leq k)$ vary in the space.

There are many solutions for $\hat{X}$, but by applying Fan's Theorem, we find that the eigenvectors associated with the smallest k eigenvalues of $\hat{Q}$ yield an optimal solution to Eq. (3). This is due to the fact that when $\hat{X}$ is composed of the k eigenvectors associated with the smallest k eigenvalues of $\hat{Q}$, we have $\hat{X}^T\hat{Q}\hat{X} = \Lambda$, where Λ is the $k \times k$ diagonal matrix composed of the smallest k eigenvalues of $\hat{Q}$. Thus, we now have an optimal solution for the quadratic assignment problem which incorporates vertex sizes. We can obtain the answer to our original problem, Eq. (2) using $X = S^{-1}\hat{X}$.

By substituting $\hat{X} = SX$ and $\hat{Q} = S^{-1^T}QS^{-1}$ into Eq. (4), we obtain an alternative view of the problem:

$$QX = MX\Lambda \quad (5)$$

which can be viewed as a *generalized eigenvalue problem* subject to the constraint $X^TMX = I$. Of course, we use Eq. (4) to solve for $\hat{X}$, and then obtain the eigenvectors X from $S^{-1}\hat{X}$.

3.1 Lower bound on the cost function

By putting together Theorem 1 with Eqs. (1),(3), and (6), we establish a lower bound on any partitioning.

$$\begin{aligned} \sum_{i=1}^{k}\lambda_i(\hat{Q}) &= \min_{\hat{X}^T\hat{X}=I}\{trace(\hat{X}^T\hat{Q}\hat{X})\} \\ &\leq trace\left(R^TQR\right) = \sum_{h=1}^{k}\frac{E_h}{||P_h||} \quad (6) \end{aligned}$$

where $\lambda_i(\hat{Q})$ denotes the i^{th} smallest eigenvalue of $\hat{Q}$. The above expression provides a tie between the continuous solution and the feasible solutions (matrices restricted to $\mathcal{D}_{n,k}$), since continuous space solutions which have a lower cost will produce a lower bound on the optimal feasible solution cost.

[1] S is the diagonal matrix with $\sqrt{m_{ii}}$ in the ii^{th} entry.

3.2 Application

The results of the modification to the eigenvalue problem presented in Eq. (4) may be used directly in *any* spectral partitioning algorithm which forms partitions from the eigenvectors of the Laplacian. The KP algorithm [4] forms k partitions by using the magnitude and orthogonality of the rows of the eigenvector matrix. MKP implements the functionality of KP, as well as supporting the use of the eigenvectors of the generalized eigenvalue problem.

4 MP Implementation

```
MP(hgraph $HG$, int $K$, int $Num\_levels$,
        algorithm Partition, boole $Refine?$)
{
    $HG_0 \leftarrow HG$
    for ($i = 1$; $i \leq Num\_levels$; $i++$)
        $HG_i \leftarrow$ Contract($HG_{i-1}$)
    $P_{Num\_levels} \leftarrow$ Partition($HG_{Num\_levels}$, $K$)
    for ($i = Num\_levels$; $i > 0$; $i--$)
    {   If (Refine? == Yes) $P_i \leftarrow$ Improve($P_i$, $K$)
        $P_{i-1} \leftarrow$ Expand($P_i$)
    }
    If ($Refine?$ == Yes) $P \leftarrow$ Improve($P_0$, $K$)
    else $P \leftarrow P_0$
    return($P$)
}
```

Figure 1: MP algorithm.

Our partitioner, MP, has been implemented in C++. MP interfaces with the LASO library by D.S. Scott, which performs the sparse matrix eigenvalue/eigenvector computation. Figure 1 illustrates how MP integrates a k-way partitioning algorithm with contraction and iterative improvement. The k-way partitioning algorithms we implemented include 1) a reimplementation of the KP partitioning algorithm [4] which uses actual vertex sizes in forming the partitions and computing the ratio-cut cost, 2) the MKP partitioning algorithm, which amounts to our new KP modified to use the eigenvectors from the generalized eigenvalue problem, and 3) an algorithm which generates a random k-way partition. The last algorithm was used to evaluate the benefit of the spectral partitioning algorithm when used in conjunction with iterative improvement methods.

4.1 Contraction and Iterative Improvement

We wanted to evaluate MP's performance in three scenarios, 1) on graphs where vertices were of non-unit size and with no hierarchical clustering, 2) on graphs with non-unit size vertices and multiple levels of contraction, 3) and finally, on graphs whose vertices were initially unit size, but became non-unit size through multiple levels of contraction. Our focus was not to find the best contraction algorithm nor the best iterative improvement algorithm, but rather to provide a framework in which to test our size-aware spectral algorithm. Other researchers have conducted more detailed studies of different contraction and improvement algorithms and their relative effects [10].

In our contraction algorithm, the edges of the hypergraph are clique expanded to obtain a graph. The algorithm orders the edges of the graph using a heap based on the weight of an edge. The $\frac{n}{2}$ edges of highest weight are removed one by one, and if either one of the two vertices of the removed edge are not already in a cluster, these vertices are merged.

Our iterative improvement algorithm is modeled after the two-way ratio-cut algorithm [12]. We have extended it to perform k-way partitioning in the following way. In turn, we select each of the k partitions as the SINK (resp. SOURCE) partition, and the remaining partitions together form the SOURCE (resp. SINK). Vertices are moved one by one from the SOURCE to the SINK based on the gain of a vertex (the total weight of the nets that would become uncut if a vertex is moved to a partition). This process is repeated until there is no more improvement. The best k-way ratio-cut solution encountered is retained. In practice only a few passes of the outer improvement loop are performed before a local minima is reached. For our experiments, we terminated the improvement step after three passes.

4.2 Implementation Issues

MP works directly with hypergraphs, only transforming the hypergraphs into graphs for the contract algorithm and eigenvector computation. Hypergraphs are converted into graphs by performing a clique expansion on the hyperedges. Each edge of the clique formed by hyperedge, e_i is given a weight of $(\frac{2}{deg(e_i)})^{\frac{3}{2}}$, as proposed in [6]. For the eigenvalue/eigenvector computations, we chose to perform clique expansions even on very large fanout nets, although in some cases it may be more practical to set an upper threshold on the nets chosen for clique expansion so that sparse matrix computations performed on the graph can be carried out efficiently. For efficiency, in the contraction algorithm, we chose to only perform clique expansion on nets of degree smaller than 100 since these nets are unlikely to affect the clustering.

5 Results

We ran experiments using seven MCNC benchmarks (number of vertices is shown in parenthesis): p1_ga(833), p2_ga(3014), t2(1663), t3(1607), t4(1515), t5(2595), and t6(1752). These benchmarks are netlists with vertex sizes. A number of different parameters for the benchmarks were run to analyze the performance of the new method. Results under the heading MKP used the solution to the generalized eigenvalue while results under KP used the eigenvectors of the Laplacian. (With unit-size vertices, no contraction and no iterative improvement this is equivalent to the KP algorithm in [4].) In order to evaluate the benefit of using the spectral information in a multilevel partitioning scheme we also used an algorithm which generates random k-way partitions. The best results out of 5 obtained from random partitions are listed under the heading RND5. Partitioning results were generated for $2, 4, 8$, and 16-way partitions. The results were reported using the scaled cost function $\frac{1}{n(k-1)} \sum_{h=1}^{k} \frac{E_h}{||P_h||}$ [4].

Tables 1 and 2 show the overall performance of the

# of levels	Iterative Improvement?	Algorithm		
		RND5	KP	MKP
0	No	43.4	4.92	2.36
0	Yes	3.01	2.61	1.38
2	No	18.0	11.4	1.91
2	Yes	2.21	1.84	1.26

Table 1: Geometric mean of scaled cost multiplied by 10^8 over all tests with actual vertex sizes.

# of levels	Iterative Improvement?	Algorithm		
		RND5	KP	MKP
0	No	163.0	17.3	17.3
0	Yes	19.3	13.6	13.6
2	No	69.2	29.0	16.2
2	Yes	13.3	14.5	12.1

Table 2: Geometric mean of scaled costs multiplied by 10^5 over all tests with unit-size vertices.

partitioner	k	p1_ga	p2_ga	t2	t3	t4	t5	t6	geom. mean
RND5	2	24.45	5.39	10.24	11.45	5.93	4.87	9.51	8.84
KP	2	13.39	5.39	12.41	9.29	5.70	5.47	7.96	7.98
MKP	2	13.39	4.58	7.96	9.29	6.40	3.31	10.34	7.19
DP-RP	2	13.5	4.58	8.07	8.98	5.78	3.12	8.21	6.78
RND5	4	17.45	7.99	12.94	13.95	8.50	5.20	11.12	10.30
KP	4	19.86	8.11	12.06	15.15	10.44	7.69	11.84	11.58
MKP	4	17.45	9.08	11.39	11.47	9.24	5.70	11.09	10.27
DP-RP	4	23.4	7.93	15.2	11.3	7.98	5.33	12.0	10.71
RND5	8	30.16	11.56	18.56	15.78	10.49	8.50	18.68	14.99
KP	8	37.59	15.34	22.58	16.29	15.61	8.38	17.77	17.45
MKP	8	29.03	10.99	20.61	15.86	14.44	7.00	16.19	14.99
DP-RP	8	33.2	11.3	21.1	15.2	12.2	7.41	17.1	15.18
RND5	16	52.91	18.94	25.95	20.59	21.42	11.95	27.86	23.37
KP	16	65.25	14.62	31.48	31.64	23.34	12.35	42.45	27.41
MKP	16	48.39	13.53	23.69	20.58	18.73	8.96	23.41	19.94
DP-RP	16	NA	NA	NA	NA	NA	NA	NA	0

Table 3: Comparing MKP with DP-RP using the scaled ratio-cut cost metric multiplied by 10^5 with unit-size vertices.

algorithms. We used the geometric mean of the results over all seven benchmarks for $k = 2, 4, 8$, and 16. In every test group, MKP gives the best answer. Table 3 lists a side-by-side comparison of the MKP partitioning results with DP-RP [1]. The results are comparable but the time complexity of MKP, $O(kn \log(n))$, is lower than DP-RP's $O(kn^2)$. A much more detailed presentation of the results is available as a technical report [13].

6 Conclusions

In this paper we have presented a modified eigenvalue formulation to account for vertex sizes, and studied its use on circuits with varying vertex sizes and within a multi-level spectral partitioning scheme. The technique is general and can be applied to any eigenvector-based partitioning algorithm.

References

[1] C. J. Alpert and A. B. Kahng. Multiway partitioning via geometric embeddings, orderings, and dynamic programming. *IEEE Trans. on CAD*, 14(11):1342–1358, Nov. 1995.

[2] S. T. Barnard and H. D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. Technical Report RNR-92-033, NASA Ames Research Center, NAS Systems Division, Moffet Field, CA, Nov. 1992.

[3] E. R. Barnes. Partitioning the nodes of a graph. *Proceedings of Graph Theory with Applications to Algorithms and Computer Science*, pages 57–72, 1985.

[4] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. on CAD*, 13(9):1088–1096, Sept. 1994.

[5] K. Fan. On a theorem of Weyl concerning eigenvalues of linear transformations. I. In *International Proceedings of the National Academy of Sciences*, volume 35, pages 652–655, 1949.

[6] J. A. Frankle. Circuit placement methods using multiple eigenvectors and linear probe techniques. Technical Report UCB/ERL M87/32, University of California, Berkeley, Electronics Research Laboratory, Berkeley, CA, May 1987.

[7] L. Hagen and A. Kahng. Fast spectral methods for ratio cut partitioning and clustering. In *Proceedings of the ICCAD*, pages 10–12, 1991.

[8] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Sciences*, 17(3):219–229, Nov. 1970.

[9] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. Technical Report SAND93-1301 * UC-405, Sandia National Laboratories, Albuquerque, New Mexico, 1993.

[10] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. Technical Report 95-035, University of Minnesota, Dept. of Computer Science, July 1995.

[11] A. Pothen, H. D. Simon, and K.-P. Lious. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal Matrix Analysis Appl.*, 11(3):430–452, July 1990.

[12] Y.-C. Wei and C.-K. Cheng. A two-level two-way partitioning algorithm. In *Proceedings of the IEEE ICCAD*, pages 516–519, 1990.

[13] J. Y. Zien, M. Schlag, and P. K. Chan. Multi-level spectral hypergraph partitioning with arbitrary vertex sizes. Technical Report UCSC-CRL-96-15, Board of Studies in Computer Engineering, University of California at Santa Cruz, July 1996.