A Graph Based Method for Generating the Fiedler Vector of Irregular Problems[1]

Michael Holzrichter[1] and Suely Oliveira[2]

[1] Texas A&M University, College Station, TX,77843-3112
[2] The University of Iowa, Iowa City, 52242, USA,
oliveira@cs.uiowa.edu,
WWW home page: http://www.cs.uiowa.edu

Abstract. In this paper we present new algorithms for spectral graph partitioning. Previously, the best partitioning methods were based on a combination of Combinatorial algorithms and application of the Lanczos method when the graph allows this method to be cheap enough. Our new algorithms are purely spectral. They calculate the Fiedler vector of the original graph and use the information about the problem in the form of a preconditioner for the graph Laplacian. In addition, we use a favorable subspace for starting the Davidson algorithm and reordering of variables for locality of memory references.

1 Introduction

Many algorithms have been developed to partition a graph into k parts such that the number of edges cut is small. This problem arises in many areas including finding fill-in reducing permutations of sparse matrices and mapping irregular data structures to nodes of a parallel computer.

Kernighan and Lin developed an effective combinatorial method based on swapping vertices [12]. Multilevel extensions of the Kernighan-Lin algorithm have proven effective for graphs with large numbers of vertices [11]. In recent years spectral approaches have received significant attention [15]. Like combinatorial methods, multilevel versions of spectral methods have proven effective for large graphs [1]. The previous multilevel spectral algorithms of [1]is based on the application of spectral algorithms at various graph levels. Previous (multilevel) combinatorial algorithms coarsen a graph until it has a small number of nodes and can be partioned more cheaply by well known techniques (which may include spectral partitioning). After a partition is found for the coarsest graph, the partition is then successively interpolated onto finer graphs and refined [10, 11]. Our focus is on finding a partition of the graph using *purely spectral techniques.* We use the graphical structure of the problem to develop a multilevel spectral partitioning algorithm applied to the original graph. Our algorithm is based on a well known algorithm for the calculation of the Fiedler vector: the

[1] This research was supported by NSF grant ASC-9528912, and currently by NSF/DARPA DMS-9874015

Davidson algorithm. The structure of the problem is incorporated in the form of a graphical preconditioner to the Davidson algorithm.

2 Spectral Methods

The graph Laplacian of graph G is $L = D - A$, where A is G's adjacency matrix and D is a diagonal matrix where $d_{i,i}$ equals to the degree of vertex v_i of the graph. One property of L is that its smallest eigenvalue is 0 and the corresponding eigenvector is $(1, 1, \ldots, 1)$. If G is connected, all other eigenvalues are greater than 0.

Fiedler [5, 6] explored the properties of the eigenvector associated with the second smallest eigenvalue. (These are now known as the "Fiedler vector" and "Fiedler value," respectively.) Spectral methods partition G based on the Fiedler vector of L. The reason why the Fiedler vector is useful for partitioning is explained in more detail in the next section.

3 Bisection as a Discrete Optimization Problem

It is well known that for any vector x

$$x^T L x = \sum_{(i,j) \in E} (x_i - x_j)^2,$$

holds (see for example Pothen et al. [15]).
Note that there is one term in the sum for each edge in the graph. Consider a vector x whose construction is based upon a partition of the graph into subgraphs P_1 and P_2. Assign $+1$ to x_i if vertex v_i is in partition P_1 and -1 if v_i is in partition P_2. Using this assignment of values to x_i and x_j, if an edge connects two vertices in the same partition then $x_i = x_j$ and the corresponding term in the Dirichlet sum will be 0. The only non-zero terms in the Dirichlet sum are those corresponding to edges with end points in separate partitions. Since the Dirichlet sum has one term for each edge and the only non-zero terms are those corresponding to edges between P_1 and P_2, it follows that $x^T L x = 4 *$ (number of edges between P_1 and P_2).

An x which minimizes the above expression corresponds to a partition which minimizes the number of edges between the partitions. The graph partitioning problem has been transformed into a discrete optimization problem with the goal of

- minimizing $\frac{1}{4} x^T L x$
- such that
  1. $e^T x = 0$ where $e = (1, 1, \ldots, 1)^T$
  2. $x^T x = n$.
  3. $x_i = \pm 1$

Condition (1) stipulates that the number of vertices in each partition be equal, and condition (2) stipulates that every vertex be assigned to one of the partitions. If we remove condition (3) the above problem can be solved using Lagrange multipliers.

We seek to minimize $f(x)$ subject to $g_1(x) = 0$ and $g_2(x) = 0$. This involves finding Lagrange multipliers λ_1 and λ_2 such that

$$\nabla f(x) - \lambda_1 \nabla g_1(x) - \lambda_2 \nabla g_2(x) = 0.$$

For this discrete optimization problem, $f(x) = \frac{1}{2} x^T L x$, $g_1(x) = e^T x$, and $g_2(x) = \frac{1}{2}(x^T x - n)$. The solution must satisfy

$$Lx - \lambda_1 e - \lambda_2 x = 0.$$

That is, $(L - \lambda_2 I)x = \lambda_1 e$. Premultiplying by e^T gives

$$e^T L x - \lambda_2 e^T x = e^T e \, \lambda_1 = n \, \lambda_1.$$

But $e^T L = 0$, $e^T x = 0$ so $\lambda_1 = 0$. Thus

$$(L - \lambda_2 I)x = 0$$

and x is an eigenvector of L.

The above development has shown how finding a partition of a graph which minimizes the number of edges cut can be transformed into an eigenvalue problem involving the graph Laplacian. This is the foundation of spectral methods. Theory exists to show the effectiveness of this approach. The work of Guattery and Miller [7] present examples of graphs for which spectral partitioning does not work well. Nevertheless, spectral partition methods work well on bounded-degree planar graphs and finite element meshes [17]. Chan et al. [3] show that spectral partitioning is optimal in the sense that the partition vector induced by it is the closest partition vector to the second eigenvector, in any l_s norm where $s \geq 1$. Nested dissection is predicated upon finding a good partition of a graph. Simon and Teng [16] examined the quality of the p-way partition produced by recursive bisection when p is a power of two. They show that recursive bisection in some cases produces a p-way partition which is much worse than the optimal p-way partition. However, if one is willing to relax the problem slightly, they show that recursive bisection will find a good partition.

4 Our Approach

We developed techniques which use a multilevel representation of a graph to accelerate the computation of its Fiedler vector. The new techniques

- provide a framework for a multilevel preconditioner
- obtain a favorable initial starting point for the eigensolver
- reorder the data structures to improve locality of memory references.

The multilevel representation of the input graph G_0 consists of a series of graphs $\{G_0, G_1, \ldots, G_n\}$ obtained by successive coarsening. Coarsening G_i is accomplished by finding a maximum matching of its vertices and then combining each pair of matched vertices to form a new vertex for G_{i+1}; unmatched vertices are replicated in G_{i+1} without modification. Connectivity of G_i is maintained by connecting two vertices in G_{i+1} with an edge if, and only if, their constituent vertices in G_i were connected by an edge. The coarsening process concludes when a graph G_n is obtained with sufficiently few vertices.

We used the Davidson algorithm [4, 2, 14] as our eigensolver. The Davidson algorithm is a subspace algorithm which iteratively builds a sequence of nested subspaces. A Rayleigh-Ritz process finds a vector in each subspace which approximates the desired eigenvector. If the Ritz vector is not sufficiently close to an eigenvector then the subspace is augmented by adding a new dimension and the process repeats. The Davidson algorithm allows the incorporation of a preconditioner. We used the structure of the graph in the development of our Davidson preconditioner. More details about this algorithm is given in [8, 9].

4.1 Multilevel Preconditioner

We will refer to our new preconditioned Davidson Algorithm as PDA. The preconditioner approximates the inverse of the graph Laplacian matrix. It operates in a manner similar to multigrid methods for solving discretizations of PDE's [13]. However, our preconditioner differs from these methods in that we do not rely on obtaining coarser problems by decimating a regular discretization. Our method works with irregular or unknown discretizations because it uses the coarse graphs for the multilevel framework. PDA can be considered as our main contribution to the current research on partitioning algorithms. The next two subsections decrib other features of our algorithms. These additional features aimed to spedd up each interation of PDA. The numerical results of Sec. 5 will shown that we were successful in improving purely spectral partitioning algorithms.

4.2 Favorable Initial Subspace

The second main strategy utilizes the multilevel representation of G_0 to construct a favorable initial subspace for the Davidson Algorithm. We call this strategy the Nested Davidson Algorithm (NDA).

The Nested Davidson Algorithm works by running the Davidson algorithm on the graphs in $\{G_0, G_1, \ldots, G_n\}$ in order from G_n to G_0. The k basis vectors spanning the subspace for G_{i+1} are used to construct l basis vectors for the initial subspace for G_i. The l new basis vectors are constructed from linear combinations of the basis vectors for G_{i+1} by interpolating them onto G_i and orthonormalizing. Once the initial basis vectors for G_i have been constructed, the Davidson algorithm is run on G_i and the process repeats for G_{i-1}.

4.3 Locality of Memory References

No assumption is made about the regularity of the input graphs. Furthermore the manner in which the coarse graphs are constructed results in data structures with irregular storage in memory. The irregular storage of data structures has the potential of reducing locality of memory accesses and thereby reducing the effectiveness of cache memories.

We developed a method called Multilevel Graph Reordering (MGR) which uses the multilevel representation of G_0 to reorder the data structures in memory to improve locality of reference. Intuitively, we permute the graph Laplacian matrices to increase the concentration of non-zero elements along the diagonal. This improved locality of reference during relaxation operations which represented a major portion of the time required to compute the Fiedler vector.

The relabeling of the graph is accomplished by imposing a tree on the vertices of the graphs $\{G_0, G_1, \ldots, G_n\}$. This tree was traversed in a depth-first manner. The vertices were relabeled in the order in which they were visited. After the relabelling was complete the data structures were rearranged in memory such that the data structures for the i^{th} vertex are stored at lower memory addresses than the data structures for the $i+1^{\text{st}}$ vertex.

An example of reordering by MGR is shown in Figures 1 and 2. The vertices are shown as well as the tree overlain on the vertices. (The edges of the graphs are not shown.) If a vertex lies to the right of another in the figure, then its datastructures occupy higher memory addresses. Notice that after reordering, the vertices with a common parent are placed next to each other in memory. This indicates that they are connected by an edge and will be referenced together during relaxation operations. Relaxation operations represented a large fraction of the total work done by our algorithms. This reordering has a positive effect of locality of reference during such operations.

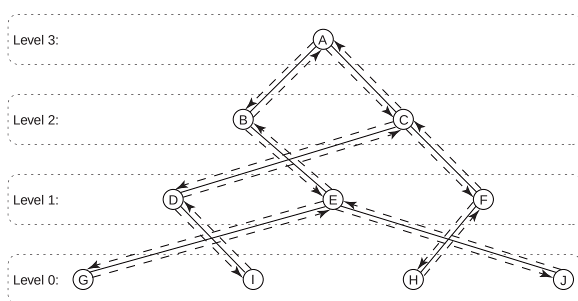

Fig. 1. Storage of data structures before reordering

The algorithms described above are complementary and can be used concurrently while computing the Fiedler vector. We call the resulting algorithm

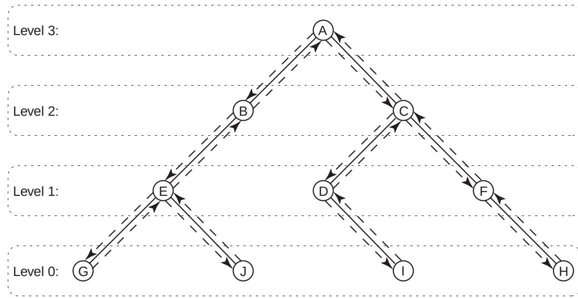

Fig. 2. Storage of data structures after reordering

when PDA, NDA and MGR are combined: the Multilevel Davidson Algorithm (MDA).

5 Numerical Results

The performance of our new algorithms were tested by computing the Fiedler vector of graphs extracted from large, sparse matrices. These matrices ranged in size from approximately $10,000 \times 10,000$ to more $200,000 \times 200,000$. The number of edges ranged from approximately 300,000 to more than 12 million. The matrices come from a variety of application domains, for example, finite element models and a matrix from financial portfolio optimization.

Figure 3 shows the ratio of the time taken by MDA when using PDA, NDA and MGR concurrently to the time taken by the Lanczos algorithm to compute the Fiedler vector. The ratio for most graphs was between 0.2 and 0.5 indicating that MDA usually computed the Fiedler vector two to five times as fast as the Lanczos algorithm. For some graphs MDA was even faster. For the **gearbox** graph MDA was only slightly faster.

Our studies indicate that the two main factors in MDA's improved performance shown in Figure 3 are the preconditioner of PDA and the favorable initial subspace provided by NDA.

We have observed that NDA does indeed find favorable subspaces. The Fiedler vector had a very small component perpendicular to the initial subspace.

We were able to measure the effect of MGR on locality of memory references. Our studies found that MGR on the average increased the number of times a primary cache line was reused by 18 percent indicating improved locality by the pattern of memory accesses.

6 Conclusions

We have developed new algorithms which utilize the multilevel representation of a graph to accelerate the computation of the Fiedler vector for that graph. The

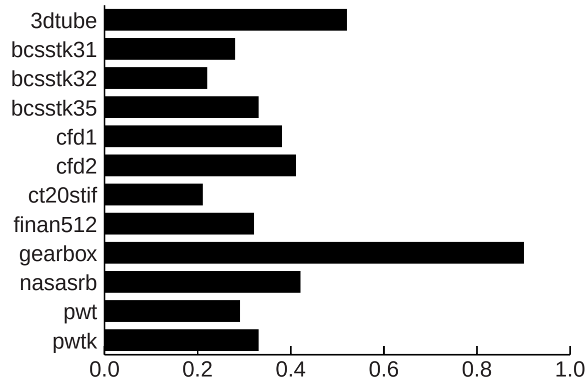

Fig. 3. Ratio of Davidson to Lanczos execution time

Preconditioned Davidson Algorithm uses the graph structure of the problem as as a framework for a multilevel preconditioner. A favorable initial subspace for the Davidson algorithm is provided by the Nested Davidson algorithm. The Multilevel Graph Reordering algorithm improves the locality of memory references by finding a permutation of the graph Laplacian matrices which concentrates the non-zero entries along the diagonal and reordering the data structures in memory accordingly. More numerical results can be found in [9]. The full algorithms for the methods described in this paper are available in [8]. PDA can be considered as our main contribution to spectral partitioning of irregular graphs.

References

1. S. Barnard and H. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. In *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, Norfolk, Virginia, 1993. SIAM, SIAM.
2. L. Borges and S. Oliveira. A parallel Davidson-type algorithm for several eigenvalues. *Journal of Computational Physics*, (144):763–770, August 1998.
3. T. Chan, P. Ciarlet Jr., and W. K. Szeto. On the optimality of the median cut spectral bisection graph partitioning method. *SIAM Journal on Computing*, 18(3):943–948, 1997.
4. E. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17:87–94, 1975.
5. M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.
6. M. Fiedler. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–632, 1975.
7. S. Guattery and G. L. Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19:701–719, 1998.

8. M. Holzrichter and S. Oliveira. New spectral graph partitioning algorithms. submitted.
9. M. Holzrichter and S. Oliveira. New graph partitioning algorithms. 1998. The University of Iowa TR-120.
10. G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular garphs. to appear in the Journal of Parallel and Distributed Computing.
11. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.
12. B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49:291–307, February 1970.
13. S. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1989.
14. S. Oliveira. A convergence proof of an iterative subspace method for eigenvalues problem. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics Selected Papers*, pages 316–325. Springer, January 1997.
15. Alex Pothen, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, 1990. Sparse matrices (Gleneden Beach, OR, 1989).
16. H. D. Simon and S. H. Teng. How good is recursive bisection. *SIAM Journal on Scientific Computing*, 18(5):1436–1445, July 1997.
17. D. Spielman and S. H. Teng. Spectral partitioning works: planar graphs and finite element meshes. In *37th Annual Symposium Foundations of Computer Science*, Burlington, Vermont, October 1996. IEEE, IEEE Press.