UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

A Sign Cut Version of the Recursive Spectral Bisection Graph Partitioning Algorithm

Tony F. Chan
W. K. Szeto

March 1994

CAM Report 94-7

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

A Sign Cut Version of the Recursive Spectral Bisection Graph Partitioning Algorithm

Tony F. Chan * W. K. Szeto †

Abstract

Recursive Spectral Bisection (RSB) is a heuristic technique for finding a minimum cut graph partition. A basic step is to compute the second eigenvector of the Laplacian of the graph and from it a bisection is obtained. The most common method is to use the *median* of the components of the second eigenvector x_2 to induce a bisection. In this short note, we propose a variant of the RSB method by using the *sign* of the components of x_2 to induce an initial partition, followed by a local exchange phase to enforce load balance. Since each phase is local in nature, it may be simpler to implement in parallel. Moreover, our experiments on several unstructured finite element meshes show that it often produces a smaller cut set than the median-cut method.

1 Introduction

One of the basic tasks in parallel computing is to partition a computational problem into smaller parts which can be mapped onto an individual processor. The objective is to have the pieces to be load balanced and the communication between the pieces as small as possible. The simplest mathematical version of this problem corresponds to finding a minimum cut equi-partition of the task graph, which is a well-known NP-complete problem.

One of the most successful heuristics proposed for approximately solving this problem is the Recursive Spectral Bisection (RSB) method first proposed by Pothen, Simon and Liou [8]. Suppose we are given an undirected, connected graph $G = (V, E)$, with $n = |V|$ even. Let L be the set of lattice vectors with components equal to ± 1, i.e. $L = \{l \in R^n | l_i \in \{\pm 1\}\}$. Let B be the set of load balanced vectors, defined as $B = \{b \in R^n | \sum_{i=1}^n b_i = 0\}$. We shall denote the set of bisection (i.e. load balanced partition) vectors by $P \equiv L \cap B$. We associate a variable x_i, with each node of the graph, which may be $+1$ or -1 corresponding the two sides of the cut. The size of the cut set corresponding to a partition vector x may then be expressed as:

$$(1) \qquad |C|_{x \in P} = \frac{1}{4} \sum_{(v,w) \in E} (x_v - x_w)^2$$

where the sum is over all edges (v, w) connecting vertices v and w of the graph.

*Department of Mathematics, Univ. of Calif. at Los Angeles, Los Angeles, CA 90024. E-mail: chan@math.ucla.edu. The work of this author was partially supported by the National Science Foundation under contract ASC 92-01266, the Army Research Office under contract DAAL03-91-G-0150, and ONR under contract ONR-N00014-92-J-1890. Part of this work was carried while the author was visiting the Computer Science Dept. at the Chinese Univ. of Hong Kong.

†Department of Computer Science, Chinese University of Hong Kong, Shatin, Hong Kong.

Now, we define the $n \times n$ Laplacian matrix $Q = (q_{ij})$ of the graph G as:

$$q_{ij} = \begin{cases} -1 & \text{if } (v_i, v_j) \in E \\ deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

It is well-known that (via summation by parts):

$$x^T Q x = \sum_{(v,w) \in E} (x_v - x_w)^2.$$

By relaxing the discreteness constraint of $x \in L$, the minimum cut set problem may be approximated as follows:

$$\min_{x \in P} |C| = \frac{1}{4} \min_{x \in P} x^T Q x \geq \frac{1}{4} \min_{x \in B, ||x||_2^2 = n} x^T Q x = \frac{1}{4} x_2^T Q x_2,$$

where λ_2 is the smallest nonzero eigenvalue of Q (which is positive semi-definite) and x_2 is the corresponding eigenvector, normalized as $||x_2||^2 = n$. As x_2 is in B (because it is orthogonal to $x_1 = (1 \cdots 1)^T$), but not in L, we need to map it to a nearby vector in P. The most common method is the median cut method which chooses the partition vector p_m by finding the median value of the components of x_2 and mapping values above the median to $+1$ and values below to -1. The partitions are then further partitioned by recursive application of the same procedure. The RSB method has been widely used in practice [10, 9], especially for unstructured finite element meshes, and further improvements have since then been proposed [1, 4, 6].

In this short note, we propose a variant of the RSB method by using the *sign* of the components of x_2 to induce an initial partition vector $p_s = sign(x_2)$. This is followed by a local exchange phase to enforce load balance. Our motivation is based on the characterizations proven in [2]:

THEOREM 1.1. *(Chan-Ciarlet-Szeto [2]) Given any $v \in R^n$, n even, let $p_m \in P$ be any median cut partition vector induced by v and p_s the corresponding sign-cut partition vector. Then, $p_m = arg\min_{p \in P} ||v - p||_\alpha$ and $p_s = arg\min_{p \in L} ||v - p||_\alpha$, where $||.||_\alpha$ denotes any l_α norm.*

That is, p_m is the closest vector in P to x_2 and p_s is the closest vector in L to x_2. We note that $p_s \notin B$ in general (i.e. not load balanced). Although p_m is *optimal* in the sense of minimizing the distance from the continous minimizer x_2 to the set of feasible partition vectors P, the mapping does not employ any information of the objective function C and thus p_m is not guaranteed to inherit the minimizing property of x_2 (otherwise we'd have solved an NP-complete problem!) On the other hand, p_s is in general closer than p_m to x_2. Our sign cut method can thus be seen as a two stage algorithm in which x_2 is first mapped onto the closest point in L, followed by a local exchange algorithm to enforce load balancing. Thus the strategy is to take two smaller steps rather than one big step and to try to make use of local information in the exchange phase to obtain a better partition.

We note that the sign-cut approach does not require a sorting phase to compute the median of x_2. In fact both the sign determination and the exchange phase are local operations which may be an advantage in a parallel implementation. Moreover, our experiments on several unstructured finite element meshes show that it has the extra bonus of often producing a smaller cut set than the median-cut method.

2 Recursive Sign-cut Spectral Bisection (RSSB)

Our implementation of the sign-cut method is very simple. After the initial sign-cut partition, we successively choose an appropriate vertex from the bigger partition and move it to the smaller partition until the two partitions are balanced. We summarize in algorithmic form the sign-cut method below:

```
function sign-cut(x_2)
np ← {i | (x_2)_i < 0}
p  ← {i | (x_2)_i ≥ 0}
d  ← ||p| - |np||
for i ← 1 to d
    v ← choice(bigger piece)
    add v to the smaller piece
end Sign-cut
```

The only algorithmic detail to be specified is how to choose a vertex from the bigger partition. Our strategy is to make the **choice** using the local information of the graph near the interface. Let S and B denote the smaller and bigger partition respectively. Let $v \in B$ be a potential vertex to be moved to S. The cut set size pre_v induced by v *before* the move is simply given by:

$$pre_v = |\{(u,v)|u \in S, (u,v) \in E\}|.$$

Similarly, the cut set size $post_v$ induced by v *after* the move is:

$$post_v = |\{(u,v)|u \in B/v, (u,v) \in E\}|.$$

Therefore, the *reduction* r_v in the size of the cut set by moving $v \in B$ to S is given by:

$$r_v = pre_v - post_v.$$

The most natural choice is to select a vertex v which maximizes r_v. However, our experience shows that simply following this strategy often produces a "fingering" effect in the shape of the boundary of the partitions. The boundary develops a "front" which protrudes into the neighboring partition. This is undersirable because the partitions will have bad aspect ratios and they can also become disconnected. To alleviate this potential problem, in the selection of points from B to be moved, we penalize those vertices which are close to already chosen ones by adding "weights" to them in a systematic way. We have found the following simple rule work well in practice. We initialize the weights of all vertices in B to zero. Then, every time a vertex v is selected from B to be moved to S, a unit weight is added to all neighbors of v. At the next iteration, we select as candidates those vertices with minimum weights. Of course, we still need a tie-breaking rule but the precise selection procedure does not seem to be critical.

3 Numerical results

We applied the sign-cut method to several unstructured finite element meshes in two dimensions. The results for three particular grids are shown in Tables 1, 2 and 3. These meshes are widely used and publicly available for testing purposes. Due to space limitation,

TABLE 1
Performance of different algorithms on the Eppstein mesh: 547 vertices

Partitions	RSB	RSB+KL	RSSB
2	47	43	41
4	98	92	89
6	177	165	157
8	304	292	285
16	456	443	439
32	665	648	645

TABLE 2
Performance of different algorithms on the Airfoil mesh: 4253 vertices

Partitions	RSB	RSB+KL	RSSB
2	138	134	92
4	231	227	197
8	402	391	333
16	631	620	563
32	1097	1082	963

we do not show them here and refer the reader to [3] for pictures of these grids. Likewise, we do not have space to show the partitions generated by the various algorithms.

In the tables, we give the cut set size for the median cut method (RSB), the median cut method with a Kernighan-Lin [7] refinement (RSB+KL) and the sign-cut method (RSSB). The Kernighan and Lin method is a classical exchange-type algorithm for the graph bisection problem. Although it is slow when used on its own, recent experiences [1, 5] show that it is very effective as a refinement procedure for RSB and therefore we include it in our tests. Since the sign-cut method already has a local exchange mechanism built in, our experience shows that a KL refinement doesn't produce much improvement and therefore we do not present those results.

These results indicate that RSSB is noticeably better than RSB and marginally better than RSB+KL.

References

[1] S. Barnard and H. Simon, *A fast multilevel implementation of recursive spectral bisection*

TABLE 3
Performance of different algorithms on the Barth mesh: 6691 vertices

Partitions	RSB	RSB+KL	RSSB
2	115	113	91
4	254	253	223
6	449	430	407
8	721	700	676
16	1185	1165	1151
32	1879	1806	1833

for partitioning unstructured problems, Tech. Rep. RNR-92-033, NASA Ames, NAS Systems Division, Applied Research Branch, NASA Ames Research Center, Mail Stop T045-1, Moffett Field, CA 94035, November 1992.

[2] T. F. Chan, P. Ciarlet, Jr., and W. Szeto, *On the optimality of the median cut spectral bisection graph partitioning method*, Tech. Rep. CAM Report 93-14, Dept. of Math, UCLA, CA 90024-1555, 1993.

[3] T. F. Chan and B. F. Smith, *Domain decomposition and multigrid methods for elliptic problems on unstructured meshes*, Tech. Rep. CAM Report 93-42, Dept. of Math, UCLA, CA 90024-1555, 1993. To appear in Proc. of 7th Int'l Conf. on Domain Decomposition Methods, Penn. State, Oct. 27-30, 1993.

[4] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations*, Tech. Rep. SAND92-1460, Sandia Nat'l Lab., Albuquerque, N.M., September 1992.

[5] ——, *The chaco user's guide version 1.0*, tech. rep., Sandia Nat'l Lab., Albuquerque, N.M., October 1993.

[6] ——, *Multidimensional spectral load balancing*, Tech. Rep. SAND93-0074, Sandia Nat'l Lab., Albuquerque, N.M., January 1993.

[7] B. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, Bell Systems Tech. Journal, 29 (1970), pp. 291–307.

[8] A. Pothen, H. Simon, and K. Liou, *Partitioning sparse matrices with eigenvector of graphs*, SIAM J. Mat. Anal. Appl., 11 (1990), pp. 430–452.

[9] H. Simon, *Partitioning of unstructured problems for parallel processing*, Computing Systems in Engineering, 2 (1991), pp. 135–148.

0] R. Williams, *Performance of dynamic load balancing algorithms for unstructured mesh calculations*, Concurrency, 3 (1991), pp. 457–481.