SIAM J. SCI. COMPUT.
Vol. 18, No. 5, pp. 1436–1445, September 1997
© 1997 Society for Industrial and Applied Mathematics
011

HOW GOOD IS RECURSIVE BISECTION?*

HORST D. SIMON† AND SHANG-HUA TENG‡

Abstract. The most commonly used p-way partitioning method is recursive bisection (RB). It first divides a graph or a mesh into two equal-sized pieces, by a "good" bisection algorithm, and then recursively divides the two pieces. Ideally, we would like to use an optimal bisection algorithm. Because the optimal bisection problem that partitions a graph into two equal-sized subgraphs to minimize the number of edges cut is NP-complete, practical RB algorithms use more efficient heuristics in place of an optimal bisection algorithm. Most such heuristics are designed to find the best possible bisection within allowed time.

We show that the RB method, even when an optimal bisection algorithm is assumed, may produce a p-way partition that is very far way from the optimal one. Our negative result is complemented by two positive ones: first we show that for some important classes of graphs that occur in practical applications, such as well-shaped finite-element and finite-difference meshes, RB is within a constant factor of the optimal one "almost always." Second, we show that if the balance condition is relaxed so that each block in the p-way partition is bounded by $2n/p$, where n is the number of vertices of the graph, then a modified RB finds an approximately balanced p-way partition whose cost is within an $O(\log p)$ factor of the cost of the optimal p-way partition.

Key words. communication cost, data and computation mapping on parallel machines, load balancing, mesh partitioning, parallel processing, recursive bisection, scalable parallel algorithms, well-shaped finite-element and finite-difference meshes

AMS subject classifications. 05C50, 68R10

PII. S1064827593255135

1. Introduction.

For a variety of applications, such as parallel scientific processing, VLSI layout, circuit testing and simulation, and sparse linear system solving, we need to partition the vertices of a graph into a given number of subsets such that the total number of edges whose endpoints are in different subsets is small [2, 5, 6, 7, 15, 16, 17, 20, 22, 23, 24, 26, 27, 29]. If p subsets are required, the problem is called the *graph p-way partition problem.* The most commonly used method for p-way partitioning, when p is a power of two, is RB. It first divides a graph into two equal-sized pieces, by a "good" bisection algorithm, and then recursively divides the two pieces. When p is not a power of two, simple variants of RB are used [8].

Ideally, we would like to use an optimal bisection algorithm in RB. However, because the *optimal bisection problem* that divides a graph into two equal-sized subgraphs to minimize the number of edges cut is NP-complete, practical RB algorithms use more efficient heuristics in place of an optimal bisection algorithm. Most such heuristics are designed to find the best possible bisection within allowed time [2, 3, 7, 24, 29, 32]. Some extended heuristics have been proposed that apply quadsection or octsection in place of bisection [18]. The published experimental results of

*Received by the editors September 9, 1993; accepted for publication (in revised form) December 6, 1995.
http://www.siam.org/journals/sisc/18-5/25513.html

†NERSC Division, Lawrence Berkeley National Laboratory, Mail Stop 50B/4230, University of California, Berkeley, CA 94720 (simon@nersc.gov). The research of this author was partially supported through NASA contract NAS 2-12961, NASA Ames Research Center, MS T27A-1, Moffett Field, CA 94035.

‡Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139. Current address: Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 (steng@cs.umn.edu). The research of this author was partially supported through NASA contract NAS 2-12961, while the author was visiting NASA Ames Research Center.

Hendrickson and Leland [18] seems to indicate that in the context of spectral partitioning quadsectioning and octsectioning, though more expensive than bisecting, give the recursive scheme better quality. Little is known, however, about how good indeed is RB even when optimal or near-optimal bisection algorithms are used, and whether more global optimization schemes should be sought.

In this paper, we show that due to its greedy nature and the lack of global information, RB may, in the worst case, produce a partition that is very far away from being optimal. In other words, optimal RB may not lead to a good p-way partition. Our results hold even for sparse graphs and more structured graphs such as planar graphs and geometric graphs [24].

On the optimistic side, our negative result is complemented by two positive results.

First, we show that for some important classes of graphs that occur in practical applications such as well-shaped finite-element and finite-difference meshes [4, 9, 24, 25, 28], RB is within a constant factor of the optimal one in the expected case. In particular, it follows from a result of Miller, Teng, Thurston, and Vavasis [24] that RB finds a p-way partition of cost $O(p^{1/d}n^{1-1/d})$ for well-shaped meshes [30] embedded in d dimensions.

Second, we show that if we relax the balance condition so that each subgraph in the partition is bounded by $2n/p$, then there exists an approximately balanced recursive partitioning algorithm (see section 5) that finds a partition whose cost is within an $O(\log p)$ factor of the cost of the optimal p-way partition. Our result implies that it may not be a good idea to insist upon the perfect bisection condition of each step of the RB scheme.

Section 2 introduces the definitions and notations that will be used in this paper. Section 3 gives a class of dense graphs and a class of sparse graphs for which optimal four-way partition has cost 12 and RB has cost $\Omega(n^2)$ and $\Omega(n)$, respectively. It also gives a tight bound of $\Theta(n^2/p^2)$ and $\Theta(n/p)$, respectively, for dense graphs and sparse graphs on the approximation ratio of RB. Section 4 shows that for well-shaped meshes in d dimensions, RB always finds a p-way partition of cost $O(p^{1/d}n^{1-1/d})$. Section 5 introduces the notion of an approximately balanced p-way partition and gives a recursive partitioning algorithm that is within an $O(\log p)$ factor of the cost of the optimal p-way partition.

2. Definitions. A *bisection* of a graph G is a division of its vertices into two disjoint subsets A and B of exactly equal sizes (for simplicity we assume that the graph has an even number of vertices). The *cost* of a bisection is the number of edges, one of whose endpoint is in A and another is in B. Similarly, a *p-way partition* of G is a division of its vertex set into p disjoint subsets of size n/p, where n is the number of vertices in G (again we assume that n is a multiple of p). The *cost* of a p-way partition is the number of edges whose endpoints are in different subsets.

When p is a power of two, the most commonly used method to find a p-way partition is to recursively apply a bisection procedure to divide the graph into p subgraphs. Assume we have a bisection function BISECTION.

Associated with RB is a tree, called the *partition tree*. Notice that the height of the partition tree is $\log p$.

If we use an optimal bisection function, the resulting RB is called an *ideal RB*. However, notice that the problem of finding an optimal bisection itself is *NP*-hard [11]. The RB scheme given above is a template of practical implementations, where we use the best available bisection algorithm. Our results can be extended to the case where BISECTION is an approximately optimal bisection algorithm.

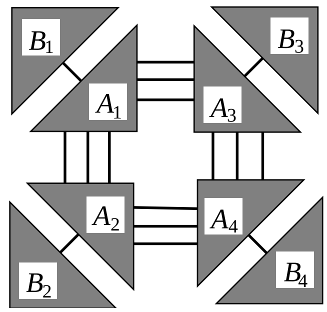

FIG. 1. *An example of four-way partitions where the optimal cost is* 12 *and ideal recursive bisection has cost* $\Omega(n^2)$ *in the dense case and* $\Omega(n)$ *in the sparse case.*

Algorithm *Recursive Bisection Scheme*(G, p)
Input: (a graph G of n vertices and an integer p, $K = n/p$.)
Output: (a p-way partition of G).

1. Apply BISECTION to find a bisection G_L and G_R of G;
2. If $|G_L| > K$ then
   - Recursive Bisection Scheme(G_L, $p/2$);
   - Recursive Bisection Scheme(G_R, $p/2$);
3. Return the subgraphs $G_1, ..., G_p$ so obtained;

For $1/2 \leq \delta \leq 1$, a δ-*bisection* (or a δ-*edge-separator*) is a partition of a graph G into two disjoint subgraphs G_L and G_R such that both $|G_L| \leq \delta|G|$ and $|G_R| \leq \delta|G|$, where the notation $|G|$ stands for the number of vertices of a graph G. The *cost* of a δ-bisection is the number of edges between G_L and G_R. By definition, a bisection is a 1/2-edge-separator.

We distinguish between two classes of graphs: dense graphs and sparse graphs. A dense graph may have $O(n^2)$-edges while a sparse graph has only $O(n)$ edges. We can further restrict that each vertex in a sparse graph has a bounded degree. As shown in [31], all well-shaped finite-element meshes in three dimensions are sparse.

A p-way partition algorithm has *approximation ratio* α, where $\alpha \geq 1$, if for each graph G it finds a p-way partition of cost at most α times the cost of an optimal p-way partition.

3. How bad can ideal recursive bisection be? In this section, we give two families of graphs, a dense family and a sparse family, that admit a constant costed p-way partition, but ideal RB produces a p-way partition of cost $\Omega(n^2/p^2)$ in the dense case and $\Omega(n/p)$ in the sparse case.

3.1. Four-way partition. We first consider four-way partitions. We consider the class of graphs given in Figure 1.

Each graph in this class has n vertices (assuming n is a power of two) and contains eight subgraphs, $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4$, where A_i has $(1/8+\epsilon_i)n$ vertices, and B_i has $(1/8-\epsilon_i)n$ vertices. The ϵ_i's $(1 \leq i \leq 4)$ satisfy the following conditions:

1. $-1/8 < \epsilon_i < 1/8$ and $\epsilon_i \neq 0$;
2. $\epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 = 0$; and
3. there is no pair of i and $j \in \{1, 2, 3, 4\}$ such that $\epsilon_i + \epsilon_j = 0$.

Such set of ϵ_i's exists. One simple way is to choose ϵ_1, ϵ_2, and ϵ_3 randomly and then choose ϵ_4 so that condition 2 holds. We can show that with high probability condition 3 holds as well.

In the dense case, A_i and B_i are *cliques*, while in the sparse case A_i and B_i are *sparse expanders*. Notice that for each constant δ such that $0 < \delta < 1$, all δ-edge-separators of a clique of n vertices have cost $\Omega(n^2)$ and all δ-edge-separators of a sparse expander of n vertices have cost $\Omega(n)$. One way to construct an expander is to choose a random bounded degree graph. It follows from a result of Erdös, Graham, and Szemerědi [10] that all δ-edge-separators of almost all such linear-sized graphs have cost $\Omega(n)$.

The optimal four-way partition divides the graph into $A_i \cup B_i$, $1 \le i \le 4$. The total cut size is 12. In contrast, ideal RB first decomposes the graph into $A_1 \cup A_2 \cup A_3 \cup A_4$ and $B_1 \cup B_2 \cup B_3 \cup B_4$. But then, because of condition 3, at least one of A_1, A_2, A_3, A_4 and one of B_1, B_2, B_3, B_4 will be divided in the next level of RB. Hence, ideal RB produces a four-way partition of cost $\Omega(n^2)$ in the dense case and $\Omega(n)$ in the sparse case.

3.2. *p*-way partitions. Our four-way example can be used to construct a tight lower bound on the approximation ratio of ideal RB for p-way partitions. We will present two classes.

In the first class, each graph contains $(p-4)$ disjoint subgraphs $G_1, ..., G_{p-4}$ of size n/p and a graph from our four-way example of size $4n/p$. Each subgraph is a clique in the dense case and a sparse expander in the sparse case.

Clearly, the optimal partition has cost 12, which is the cost of decomposing the subgraph from the four-way example. In the first $(\log p - 2)$ levels, ideal RB simply decomposes the graph into blocks of four subgraphs each. The four-way example is assigned as one block. The cost so far is zero. To complete the p-way partition, ideal RB has to divide the four-way example into four subgraphs. Hence, it yields a p-way partition of cost $\Omega(n^2/p^2)$ in the dense case and $\Omega(n/p)$ in the sparse case.

In the second class, each graph contains $p/4$ disjoint subgraphs from our four-way example of size $4n/p$. The optimal p-partition has cost $3p$, while the partition of ideal RB has cost $\Omega(n^2/p)$ in the dense case and $\Omega(n)$ in the sparse case.

Interestingly, on both classes, the approximation ratio of ideal RB is $\Omega(n^2/p^2)$ in the dense case and $\Omega(n/p)$ in the sparse case. Does RB always achieve approximation ratios of these orders, respectively? We now show that the answer is "yes."

LEMMA 3.1. *Ideal RB has worst-case approximation ratio of* $\Theta(n^2/p^2)$ *in the dense case and* $\Theta(n/p)$ *in the sparse case for p-way partitions.*

Proof. The two classes of graphs presented in this subsection provide the lower bound on the approximation ratio of ideal RB. So we only need to prove the upper bound. We give a proof for the sparse case only. The proof for the dense case is essentially the same.

The cost of the p-way partition of ideal RB is bounded from above by the total number of edges in the graph. Hence, by our sparsity assumption, it is $O(n)$. Thus, if the cost of the optimal p-way partition is $\Omega(p)$, then the recursive bisection has approximation ratio $O(n/p)$ because it has cost at most $O(n)$.

To complete the proof, we assume that the cost of the optimal p-way partition is $k < p$. Let $C_1, C_2, ..., C_p$ be the subgraphs in an optimal p-partition. Note that the number of subgraphs that are connected to some other subgraphs is at most k. We call them the *connecting blocks*. Ideal RB has zero cost except when the subgraph containing those connecting blocks. Because we have at most k connecting blocks and the total number of edges in connecting blocks is at most $O(kn/p)$, this implies that ideal RB generates a p-way partition of cost at most $O(kn/p)$. Since the cost of the optimal p-way partition has cost k, the approximation ratio is again $O(n/p)$. ∎

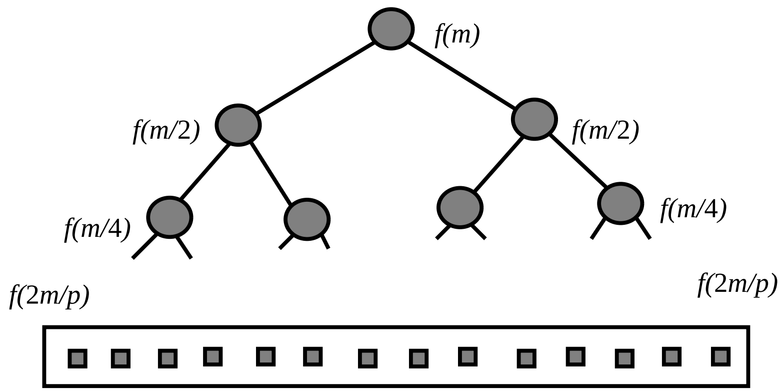

FIG. 2. *The partition tree generated by RB and the upper bound of its cost.*

4. Graphs with a family of small edge bisectors. Many graphs from practical applications have a family of *small separators* [24]; i.e., they have the property that each of its subgraphs has a bisector of size sublinear in the number of vertices of the subgraph. Formally, for a given integer function f, a graph G has a *family of f-bisectors* if each of its subgraphs of s vertices has a bisector of size bounded by $f(s)$. How good is RB on the class of graphs that has a family of f-bisectors?

LEMMA 4.1. *If G has a family of f-bisectors, then RB finds a p-way partition of cost*

$$\sum_{i}^{\log p-1} 2^i f(n/2^i).$$

Proof. Figure 2 illustrates the partition tree of the p-way partition given by RB and the upper bound on the cost of the bisectors at each level of partition tree.

Notice that the partition tree has $(\log p - 1)$ levels of internal nodes. The total cost of the ith level (where the root is at level 0) is no more than $2^i f(n/2^i)$. Hence, the total cost of the p-way partition generated by RB is at most

$$\sum_{i}^{\log p-1} 2^i f(n/2^i). \qquad \square$$

In theory as well as in practice, the condition of a family of f-bisectors can be relaxed to the condition of a *family of f-separators*; that is, every subgraph of G of s vertices has a δ-bisection, for some constant δ, of cost bounded by $f(s)$. Lipton and Tarjan [21] and Gilbert and Tarjan [14] showed that if a graph has a family of f-separators, then it has a family of $O(f)$-bisectors. The following are some classes of graphs that have a family of small separators and hence have a family of small bisectors as well.

- Planar graphs [21] have a family of $O(\sqrt{n})$-separators.
- Bounded genus graphs [12] have a family of $O(\sqrt{gn})$-separators, where g is the genus of the graphs.
- Bounded minor graphs [1] have a family of $O(h^{1.5}\sqrt{n})$-separators, where h is the size of the largest minor clique. A *minor* of a graph is a subgraph obtained from the original graph by contracting edges. For example, no planar graph has a minor isomorphic to a five-clique.

- Well-shaped meshes [24] have a family of $O(n^{1-1/d})$-separators, where d is the dimension of the space in which the meshes are embedded.
- k-nearest neighborhood graphs [24] have a family of $O(k^{1/d}n^{1-1/d})$-separators.

The following lemma follows directly from Lemma 4.1.

LEMMA 4.2. *If $f(n) = n^{1-1/d}$, then RB finds a p-way partition of cost $O(p^{1/d}n^{1-1/d})$.*

Proof. The cost of the p-way partition constructed by RB is bounded from above by

$$\begin{aligned} &\sum_{i}^{\log p-1} 2^i (n/2^i)^{1-1/d} \\ &= n^{1-1/d}\left(\sum_{i=1}^{\log p-1} 2^{i/d}\right) \\ &= O(p^{1/d}n^{1-1/d}). \qquad \square \end{aligned}$$

Most well-shaped meshes (in d dimensions) in practical applications have no p-way partition of size $o(p^{1/d}n^{1-1/d})$ [31]. For example, a d-dimensional regular grid has no p-way partition of size $o(p^{1/d}n^{1-1/d})$. So the p-way partition of RB is optimal (up to a constant factor).

Notice that we did not require the RB scheme to find optimal bisections in order to achieve the result of this section. All we require is that the RB scheme should use a bisection algorithm that finds an f-bisector in a graph that has a family of f-bisectors.

We can extend the lower-bound argument of section 3 to graphs discussed in this section.

THEOREM 4.3. *Ideal RB has worst-case approximation ratio of $\Theta(\sqrt{n/p})$ for planar graphs and $\Theta((n/p)^{1-1/d})$ for well-shaped meshes in d dimensions.*

5. Approximately balanced p-way partition. We observe that even though we use general edge separators (not necessarily bisection [13, 21, 24]) or use *minimum quotient separator* of [19] at each level of a recursive partitioning scheme, we still can not guarantee the approximation ratio as long as the final partition is required to be a (perfectly balanced) p-way partition. All of the results in the previous sections generalize. *Can we trade the balance condition for a better approximation ratio of a recursive partitioning scheme?*

Let $\beta \geq 1$ be a real number. A (β, p)*-way partition* decomposes G into disjoint $G_1, ..., G_p$ such that $|G_i| \leq \beta|G|/p$ for all $1 \leq i \leq p$. Thus, a p-way partition is a $(1, p)$-way partition. The *cost* of a (β, p)-way partition is the number of edges of G whose two endpoints are in different subgraphs.

Recall that a δ-bisection, $1/2 \leq \delta < 1$, divides G into two disjoint subgraphs G_L and G_R such that both $|G_L| \leq \delta|G|$ and $|G_R| \leq \delta|G|$. The cost of a δ-bisection is the number of edges between G_L and G_R.

We now give a recursive partitioning scheme. We use an APPROXIMATE BISECTION function to divide the graph into two disjoint subgraphs.

Notice that the procedure recursive cutting above may return less than p subgraphs (see case $h < p$) because the recursion terminates once the size of the subgraph is no more than $2n/p$. In this case, we simply assume that other $p - h$ subgraphs are empty.

We now show that if we use an APPROXIMATE BISECTION that finds an optimal $(1/2 + 1/s)$-bisection of G, then the recursive partition scheme finds a $(2, p)$-

way partition whose cost is at most $O(\log p)$ times the cost of the optimal p-way partition. We then extend the result to the case when an approximately optimal $(1/2+1/s)$-bisection algorithm is used.

Algorithm (*Recursive Partitioning Scheme*)
Input: (a graph G of n vertices and an integer p).
Output: (a $(2,p)$-way partition of G).

1. Let $K = n/p$
2. Let $G_1, ..., G_h$ be the h subgraphs obtained from running the subroutine Recursive Cutting(G,K) below;
3. If $h \leq p$, then return $(G_1, ..., G_h)$ else repeatedly merge the smallest two subgraphs until p subgraphs remain.

Subroutine (*Recursive Cutting*(G, K))

1. Let $s = |G|/K$;
2. Apply APPROXIMATE BISECTION to find a $(1/2+1/s)$-bisection G_L and G_R of G;
3. If $|G_L| > 2K$ then Recursive Cutting(G_L,K);
4. If $|G_R| > 2K$ then Recursive Cutting(G_R, K);

We first prove a simple lemma that is useful for our main result.

LEMMA 5.1. *Suppose $X = \{x_1, ..., x_m\}$ is a set of positive reals such that $0 \leq x_i \leq 1$. Then X can be divided into two subsets X_1 and X_2 such that*

$$\left| \sum_{x \in X_1} x - \sum_{y \in X_2} y \right| \leq 1.$$

Proof. We can construct the two subsets by a greedy approach: first we put all elements from X in a queue and maintain two sets that are initially empty. Then we assign the largest element from the queue to the set with smaller total sum (and of course delete the chosen element from the queue). We repeat this process until the queue is empty. Since all elements from X are no more than one, the sums of the two sets so constructed differ by the value of at most one element, which is at most one. ∎

THEOREM 5.2. *Let G be a graph and p be a positive integer. If the cost of the optimal p-way partition of G is C, then the recursive partitioning scheme that uses an optimal $(1/2+1/s)$-bisection algorithm finds a $(2,p)$-way partition of cost at most $O(C \log p)$.*

Proof. The basic idea of the proof is to argue that the cost induced at each level of the partitioning tree is at most C. Because the partitioning tree has $O(\log p)$ levels, the theorem then follows. Without loss of generality, we assume that p is a power of two.

Clearly, the partition associated with the root of the partitioning tree has cost at most C. This can be shown by the following argument which is more complex than needed but useful for bounding the cost of other levels of the partition tree.

Let $B_1, ..., B_p$ be the p subgraphs of an optimal p-way partition of G. We call an edge a *bridging edge* if its endpoints are in different subgraphs from $B_1, ..., B_p$. After removing all the bridging edges, we can group $B_1, ..., B_p$ into two subsets of equal size. This implies that G has a $(1/2+1/p)$-bisection, in fact a perfect bisection, of cost at most C.

The size of B_i, $|B_i|$, is equal to $K = n/p$. We now show that there exists an approximately balanced bisection for each node at level i so that the total cost of level i is at most C. Because we assume that the recursive partitioning scheme uses an optimal $(1/2 + 1/s)$-bisection algorithm for each node, the total cost of bisectors generated at level i can only be smaller, and hence is no more than C.

Implicitly, $B_1, ..., B_p$ themselves may be decomposed into pieces after the top $i-1$ levels of the partition tree. Each node of the partitioning tree has a subgraph that is formed by a subset of these pieces of B_i's. Notice that the size of those pieces is at most K. Now imagine that we delete all the bridging edges which are not removed in the top $(i-1)$ levels of the partition tree. The total cost of these bridging edges is at most C because these edges connect $B_1, ..., B_p$ of the optimal p-way partition. After removing these bridging edges, each node at level i of the partition tree contains a subset of the pieces of B_i's, whose size is at most K.

We now apply Lemma 5.1 to divide the pieces of each node into two groups. By Lemma 5.1, if the subgraph of a node at level i has size sK, then the larger group has size at most $sK/2 + K$. Hence, the grouping gives a $(1/2 + 1/s)$-bisection.

After deleting the bridging edges, we did not remove any other edges, and hence the total cost of bisectors at level i is at most C, completing the proof. $\square$

We can extend Theorem 5.2 to more practical case, where an approximately optimal $(1/2 + 1/s)$-bisection algorithm is used in the recursive partitioning scheme. The argument is very similar.

THEOREM 5.3. *Let G be a graph and p be a positive integer. If the cost of the optimal p-way partition of G is C, then the recursive partitioning scheme that uses a $(1/2+1/s)$-bisection algorithm with approximation ratio α finds a $(2, p)$-way partition of cost at most $O(\alpha C \log p)$.*

In practice, we can use the best available approximate bisection algorithms, such as the spectral algorithm [18, 29] and the geometric algorithm [24]. We can use Theorem 5.3 to justify their performance. Theoretically, we can apply Theorem 5.3 in conjunction with the following result of Leighton and Rao [19] to obtain the first provably good approximately balanced p-way partition algorithm.

THEOREM 5.4 (Leighton and Rao). *Let δ_0 and δ be two constants such that $1/2 < \delta_0 < \delta < 1$. Let G be a graph such that G has a δ_0-bisection of cost C. There is a polynomial time algorithm, using multicommodity flow, to find a δ-bisection of G of cost at most $O(C \log n)$, where n is the number of vertices of G and the constant hidden in big-O depends only on δ and δ_0.*

COROLLARY 5.1. *There is a polynomial time algorithm that finds a $(2, p)$-way partition of cost $O(C \log n \log p)$, where C is the cost of the optimal p-way partition.*

If we choose $K = \epsilon n/p$ in step 1 of the recursive partitioning scheme, then we have the following strengthened result. The proof is similar to that of Theorem 5.3.

THEOREM 5.5. *Let G be a graph and p be a positive integer. If the cost of the optimal (p/ϵ)-way partition of G is C, then the modified recursive partitioning scheme that uses an optimal $(1/2 + 1/s)$-bisection algorithm finds a $(1 + \epsilon, p)$-way partition of cost at most $O(C \log p)$. The modified recursive partitioning scheme that uses a $(1/2 + 1/s)$-bisection algorithm with approximation ratio α finds a $(1 + \epsilon, p)$-way partition of cost at most $O(\alpha C \log p)$. Therefore, there exists a polynomial time algorithm that finds a $(1 + \epsilon, p)$-way partition of cost $O(C \log n \log p)$.*

6. Final remarks. We address a fundamental issue in graph partitioning. Our results of section 3 can be extended to the case when recursive quadsectioning or octsectioning is used, providing some theoretical evidence to the experimental claim that recursive quadsectioning and octsectioning usually find a better p-way partition. The result in section 4 is mainly observational and follows quite directly from the previous separator results [1, 12, 21, 24]. These results give an absolute upper bound on the cut-size of the p-way partition. It shows that the ratio of cut-size to the graph size is $O((p/n)^{1/d}) < 1$. So the ratio of computations to communication in processing well-shaped meshes is reasonably balanced as p and especially n increase, demonstrating that the partitioning-based parallel algorithms are scalable. The result of section 5 gives a theoretical justification to the recursive approach taken in [7, 13, 24, 31] and many similar heuristics currently implemented. We expect to see these ideas extended for better, perhaps more global, schemes for approximating p-way partitioning.

Acknowledgments. We would like to thank David Bailey and Jim Ruppert for their careful proofreading of the draft and helpful comments.

REFERENCES

[1] N. Alon, P. Seymour, and R. Thomas, *A separator theorem for graphs with an excluded minor and its applications*, in ACM STOC, Baltimore, MD, 1990, pp. 293–299.
[2] S. T. Barnard and H. D. Simon, *A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems*, Concurrency: Practice and Experience, 6(2) 1994, pp. 101–117.
[3] M. J. Berger and S. Bokhari, *A partitioning strategy for nonuniform problems on multiprocessors*, IEEE Trans. Comput., C-36 (1987), pp. 570–580.
[4] M. Bern, D. Eppstein, and J. R. Gilbert, *Provably good mesh generation*, in 31st Annual Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1990, pp. 231–241.
[5] H. Berryman, J. Saltz, and J. Scroggs, *Execution time support for adaptive scientific algorithms on distributed memory machines*, Concurrency: Practice and Experience, 3(3) (1991), pp. 159–178.
[6] P. E. Bjørstad and O. B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., 23 (1986), pp. 1097–1120.
[7] G. E. Blelloch, A. Feldmann, O. Ghattas, J. R. Gilbert, G. L. Miller, D. R. O'Hallaron, E. J. Schwabe, J. R. Shewchuk, and S.-H. Teng, *Automated parallel solution of unstructured PDE problems*, CACM, 1998, to appear.
[8] P. K. Chan, M. Schlag, and J. Zien, *Spectral k-way ratio cut partitioning and clustering*, in Proc. Symposium on Integrated Systems, Seattle, WA, 1993.
[9] L. P. Chew, *Guaranteed Quality Triangular Meshes*, Tech. report, TR 89-893, Department of Computer Science, Cornell University, Ithaca, NY, 1989.
[10] P. Erdós, R. L. Graham, and E. Szemerédi, *On sparse graphs with dense long paths*, Comput. Math. Appl., 1 (1975), pp. 365–369.
[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA, 1979.
[12] J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan, *A separator theorem for graphs of bounded genus*, J. Algorithms, 5 (1984), pp. 391–407.
[13] J. R. Gilbert, G. L. Miller, and S.-H. Teng, *Geometric mesh partitioning: Implementation and experiments*, SIAM J. Sci. Comput., to appear.
[14] J. R. Gilbert and R. E. Tarjan, *The analysis of a nested dissection algorithm*, Numer. Math., 50 (1987), pp. 377–404.
[15] J. A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
[16] J. A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.
[17] S. Hammond and R. Schreiber, *Solving Unstructured Grid Problems on Massively Parallel Computers*, Tech. report TR 90.22, RIACS, Mountain View, CA, 1990.
[18] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations*, SIAM J. Sci. Comput., 16 (1995), pp. 452–469.
[19] F. T. Leighton and S. Rao, *An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms*, in 29th Annual Symposium on Foundations of Computer Science, White Plains, NY, 1988, pp. 422–431.

[20] C. E. Leiserson, *Area efficient VLSI computation*, in Foundations of Computing, MIT Press, Cambridge, MA, 1983.

[21] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[22] R. J. Lipton, D. J. Rose, and R. E. Tarjan, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.

[23] J. W. H. Liu, *The solution of mesh equations on a parallel computer*, in 2nd Langley Conference on Scientific Computing, NASA, Langley, VA, 1974, pp. 17–24.

[24] G. L. Miller, S.-H. Teng, W. Thurston, and S.A. Vavasis, *Automatic mesh partitioning*, in Sparse Matrix Computations: Graph Theory Issues and Algorithms, A. George, J. Gilbert, and J. Liu, eds., IMA Vol. Math. Appl., Springer-Verlag, New York, 1993, pp. 57–84.

[25] S. A. Mitchell and S. A. Vavasis, *Quality mesh generation in three dimensions*, in Proc. ACM Symposium on Computational Geometry, ACM, Berlin, 1992, pp. 212–221.

[26] V. Pan and J. Reif, *Efficient parallel solution of linear systems*, in Proc. 17th Annual ACM Symposium on Theory of Computing, Providence, RI, May 1985, pp. 143–152.

[27] A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

[28] J. Ruppert, *Results on Triangulation and High Quality Mesh Generation*, Ph.D. thesis, University of California at Berkeley, Berkeley, CA, 1992.

[29] H. D. Simon, *Partitioning of unstructured problems for parallel processing*, Comput. Systems Engrg., 2 (1991), pp. 135–148.

[30] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[31] S.-H. Teng, *Points, Spheres, and Separators: A Unified Geometric Approach to Graph Partitioning*, Ph.D. thesis, CMU-CS-91-184, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

[32] R. D. Williams, *Performance of dynamic load balancing algorithms for unstructured mesh calculations*, Concurrency, 3 (1991), pp. 457–469.