Min-Max-Boundary Domain Decomposition*

Marcos Kiwi† Daniel A. Spielman‡ Shang-Hua Teng§

Abstract

Domain decomposition is one of the most effective and popular parallel computing techniques for solving large scale numerical systems. In the special case when the amount of computation in a subdomain is proportional to the volume of the subdomain, domain decomposition amounts to minimizing the surface area of each subdomain while dividing the volume evenly. Motivated by this fact, we study the following *min–max boundary* multi–way partitioning problem: Given a graph G and an integer $k > 1$, we would like to divide G into k subgraphs $G_1, \ldots, G_k$ (by removing edges) such that (i) $|G_i| = \Theta(|G|/k)$ for all $i \in \{1, \ldots, k\}$; and (ii) the maximum boundary size of any subgraph (the set of edges connecting it with other subgraphs) is minimized.

We provide an algorithm that given G, a well–shaped mesh in d dimensions, finds a partition of G into k subgraphs $G_1, \ldots, G_k$, such that for all i, G_i has $\Theta(|G|/k)$ vertices and the number of edges connecting G_i with the other subgraphs is $O((|G|/k)^{1-1/d})$. Our algorithm can find such a partition in $O(|G| \log k)$ time. Finally, we extend our results to vertex–weighted and vertex–based graph decomposition. Our results can be used to simultaneously balance the computational and memory loads on a distributed–memory parallel computer without incurring large communication overhead.

1 Introduction

Domain decomposition is one of the most effective and popular divide–and–conquer techniques for solving large scale numerical systems on parallel computers [7, 9]. Using domain

*Part of this work was done while Daniel Spielman and Shang-Hua Teng were visiting U. Chile.

†Departamento de Ingeniería Matemática, Fac. de Cs. Físicas y Matemáticas, U. Chile, Santiago 170–3, Correo 3, Chile. mkiwi@dim.uchile.cl, www.dim.uchile.cl/~mkiwi. Supported in part by Fondecyt No. 1981182 and Fondap in Applied Mathematics 1998.

‡Department of Mathematics, M.I.T., Cambridge, MA 02139, USA. spielman@math.mit.edu, www-math.mit.edu/~spielman. Supported in part by an NSF CAREER award (CCR-9701304) and an Alfred P. Sloan Research Fellowship.

§Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. steng@cs.uiuc.edu, www.cs.umn.edu/~steng. Supported in part by an NSF CAREER award (CCR-9502540), an Alfred P. Sloan Research Fellowship, and an Intel research grant. Part of the work was done while at University of Minnesota.

decomposition, one divides the domain of a problem into subdomains so that the solutions to induced subproblems on the subdomains can be efficiently combined to solve the original problem on the whole domain. When applying domain decomposition to the solution of partial differential equations, it is desirable to decompose the domain into subdomains in such a way that each induced subproblem requires approximately the same amount of computation and the communication among the subdomains is minimized [7].

We first focus on the special case in which the amount of computational work associated with a subdomain is proportional to its volume. In this case, domain decomposition amounts to dividing the volume among the subdomains equally while minimizing the boundary/surface area of each. The ratio of the measure of the boundary to the measure of the computational work of a subdomain is sometimes referred to as the *surface–to–volume ratio* or the *communication–to–computation ratio* of the subdomain. Minimizing this ratio plays a key role in efficient parallel iterative methods [9].

To solve partial differential equations numerically, one discretizes the domain into a mesh of well–shaped elements such as simplices or hexahedral elements. As the density of mesh points, and hence the size of mesh elements, may vary within the domain, one may obtain an unstructured mesh [5, 14, 18]. Obtaining good partitions of unstructured meshes is, in general, significantly more challenging than partitioning their uniform/regular counterparts.

The main result established in this work is that every d-dimensional well–shaped unstructured mesh has a k–way partition in which the surface–to–volume ratio of every sub–mesh is almost as small as that of a regular d–dimensional grid that has the same number of nodes.

In Section 2, we introduce the problem of min–max–boundary multi–way partitioning. In Section 3, we describe a multi–way partitioning algorithm and present our main result. In Section 4, we extend the results of Section 3 to graphs with non–negative weights at each vertex. More precisely, we propose an efficient algorithm that partitions vertex–weighted graphs into subgraphs of similar total weight and vertex size while maintaining low surface–to–volume ratio in each subgraph. Such multi–way partitioning algorithms can be used to simultaneously balance the computational work and the memory requirements on a distributed–memory parallel computer while keeping communication overhead low. In Section 5, we present some experimental results and discuss the vertex–based partitioning problem.

2 Multi–way Partitioning

A *bisection* of a graph G is a division of its vertices into two disjoint subsets whose sizes differ by at most one. The *cut–size* of a bisection is the number of edges with endpoints in both subsets. In general, for every integer $k > 1$, a *k-way partition* of G is a division of its vertex set into k disjoint subsets of size $\lceil |G|/k \rceil$ or $\lfloor |G|/k \rfloor$, where $|G|$ denotes the number of vertices in G.

Partitions that evenly divide the vertices are not necessary in most applications [16]. In most cases, *balanced partitions* suffice. Given a graph $G = (V, E)$, an integer $k > 1$ and a real number $\beta \geq 1$, a partition $P = \{G_1, \ldots, G_k\}$ is a (β, k)-partition of G if $|V_i| \leq \beta \lceil |G|/k \rceil$, for all $i \in \{1, \ldots, k\}$, where V_i is the vertex set of G_i. We denote by $\partial_V(G_i)$ the set of *boundary–vertices* of G_i—the set of vertices in V_i that are connected by an edge of G to a vertex not in V_i—and by $\partial_E(G_i)$ the *boundary–edges* of G_i—the set of edges in G exactly one of whose endpoints is in V_i.

We consider the following two costs associated with a (β, k)-partition:

$$\text{total-boundary}_E(P) = \left(\sum_{i=1}^{k} |\partial_E(G_i)| \right) /2$$

$$\text{max-boundary}_E(P) = \max_{i=1,\ldots,k} |\partial_E(G_i)| \, .$$

The problem of *min–total–boundary (multi–way) partitioning* is to construct a (β, k)-partition that minimizes total-boundary, while *min–max–boundary (multi–way) partitioning* is to construct a (β, k)-partition that minimizes max-boundary.

3 Bounds for Min–Max–Boundary Partitioning

We first introduce some terminology. Let $\mathcal{G}$ be a family of graphs that is closed under the subgraph operation (i.e., every subgraph of every graph in the family is also in the family). For $0 < \alpha < 1$, we say $\mathcal{G}$ *has an n^α-separator theorem* or $\mathcal{G}$ is *n^α-separable* if there is a constant c such that every n-node graph in $\mathcal{G}$ has a bisection of cut–size at most cn^α. Moreover, we refer to the latter type of bisections as n^α-separators. For example, Lipton and Tarjan [12] showed that bounded–degree planar graphs are $n^{1/2}$-separable; Gilbert, Hutchinson, and Tarjan [11] showed that bounded–degree graphs with bounded genus are $n^{1/2}$-separable; Alon, Seymour, and Thomas [1] proved that bounded–degree graphs that do not have an h-clique minor for a constant h are $n^{1/2}$-separable. Miller, Teng, Thurston, and Vavasis [14, 15, 17] showed that well–shaped meshes in $\mathbb{R}^d$ and nearest neighbor graphs in $\mathbb{R}^d$ are $(n^{1-1/d})$–separable. (More information concerning small separators can be found in [12, 17].) We denote by $\mathcal{G}(\alpha)$ a family of graphs that is n^α-separable and closed under the subgraph operation.

The min–total–boundary partitioning problem was addressed by Simon and Teng [16], who showed:

Lemma 3.1 *Let k be an integer such that $k > 1$. Then, for every bounded degree graph G in $\mathcal{G}(\alpha)$ a k-way partition P such that* $\text{total-boundary}_E(P) = O(k^{1-\alpha}|G|^\alpha)$ *can be constructed.*

A related problem is that of finding bifurcators [6]. A graph G has an $(F_0, F_1, \ldots, F_r)$-*decomposition tree* if G can be decomposed into two subgraphs G_0 and G_1 by removing no more than F_0 edges from G, and in turn, both G_0 and G_1 can be decomposed into smaller subgraphs by removing no more than F_1 edges from each, and so on. An n-node graph has a *β-bifurcator* of size F if it has an $(F, F/\beta, F/\beta^2, \ldots, 1)$-decomposition tree. Bhatt and

Leighton [6] showed that every graph in $\mathcal{G}(\alpha)$ has a $\sqrt{2}$-bifurcator of size $O(\sqrt{n})$ if $\alpha \leq 1/2$, and has a $\sqrt{2}$-bifurcator of size $O(n^\alpha)$ if $\alpha > 1/2$.

The following theorem concerning min–max–boundary partitioning is the main result of this section.

Theorem 3.2 *Let k be an integer such that $k > 1$ and $\psi \cdot (1 - 2^{1-1/\alpha}) > 4k/|G|$. Let G be a bounded–degree graph in $\mathcal{G}(\alpha)$. Then, one can construct a $(1+\psi, k)$-partition P of G such that* $\text{max-boundary}_E(P) = O(\frac{1}{\psi^{1-\alpha}}(|G|/k)^\alpha)$.

Notice that Theorem 3.2 essentially implies Lemma 3.1. Thus, our main result can be seen as an extension of the result of [16] cited above.

3.1 Simultaneous Partition of Vertices and Boundary

We first examine a simple example. Consider a $\sqrt{n} \times \sqrt{n}$ grid in two dimensions where we assume both k and $\sqrt{n}$ are powers of two. One way of partitioning the grid is to divide it into two $\sqrt{n} \times \sqrt{n}/2$ grids by removing the edge in the middle of every row (a $\sqrt{n}$-separator), and then divide each of the two sub–grids into two $\sqrt{n}/2 \times \sqrt{n}/2$ sub–grids by removing the middle edge of every column. This process can be continued by recursively dividing the sub–grids until k disconnected sub–grids are found. Clearly, each final sub–grid has n/k vertices and at most $4\sqrt{n/k}$ boundary–edges. However, the naive recursive application of the separator theorem of Lipton and Tarjan does not, in general, guarantee the generation of a k-way partition P with $\text{max-boundary}_E(P) = O(\sqrt{n/k})$ for all bounded degree n-node planar graphs. The following somewhat stronger version of the small–separator theorem was used in partitioning the 2D grid: at every stage of the divide–and–conquer,

(1) Each subgraph was divided into two subgraphs of the same size by removing a set of edges whose size is on the order of the square–root of the size of the subgraph (*a la* the standard Lipton–Tarjan Theorem).

(2) The boundary–vertices of the subgraphs were divided evenly.

Our method is motivated by the latter observation, more formally given below.

Lemma 3.3 *Let $k > 1$ be a power of two. Let G be a bounded–degree graph in $\mathcal{G}(\alpha)$ such that $|G|$ is a power of two. If in every stage of a divide–and–conquer partitioning procedure the vertices and boundary–vertices of each subgraph are evenly divided by a separator whose size is on the order of the α-th power of the size of the subgraph, then the divide–and–conquer procedure, on input G, will generate a k-way partition whose* max-boundary *is $O((|G|/k)^\alpha)$.*

Proof: Let $s(i)$ be the maximum possible number of boundary–vertices for graphs at level i of the divide–and–conquer partitioning procedure. It follows from the assumption of the lemma that there exists a constant c such that $s(1) \leq c(|G|/2)^\alpha$, and if $i \geq 1$,

$$\begin{aligned} s(i) &\leq s(i-1)/2 + c \cdot (|G|/2^i)^\alpha \\ &\leq c \cdot (|G|/2^i)^\alpha \left(\sum_{j=0}^{i-1} 2^{j(\alpha-1)} \right) . \end{aligned}$$

Since $\alpha < 1$, we get that $s(i) = O((|G|/2^i)^\alpha)$. Fixing $i = \log k$, we have $s(i) = O((|G|/k)^\alpha)$. The lemma follows from the assumption that G is a bounded–degree graph. □

Unfortunately, we may not always be able to find a small separator that evenly divides both vertices and boundary–vertices. We show that this simultaneous partition can be achieved approximately.

A variation of the following lemma was proved by Lipton and Tarjan [12].

Lemma 3.4 *Let $G = (V, E)$ be a graph in $\mathcal{G}(\alpha)$. Let S be a subset of V. Then, one can find an $O(|G|^\alpha)$-separator that divides G into two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|S \cap V_1| = \lfloor |S|/2 \rfloor$ and $|S \cap V_2| = \lceil |S|/2 \rceil$.*

For completeness we present the algorithm and proof here.

1. Let $D^{(0)} = G$, $V_1 = V_2 = \emptyset$, and $i = 0$.
2. Repeat until $D^{(i)}$ is an empty graph,
   (a) If $|D^{(i)}| = 1$, then let $F^{(i)} = D^{(i)}$ and $\bar{F}^{(i)} = \emptyset$; otherwise find a bisection of cut–size $O(|D^{(i)}|^\alpha)$ that divides the vertex set of $D^{(i)}$ into $F^{(i)}$ and $\bar{F}^{(i)}$ and assume, without loss of generality that $|S \cap F^{(i)}| \le |S \cap \bar{F}^{(i)}|$.
   (b) If $|S \cap V_1| \le |S \cap V_2|$, let $V_1 = V_1 \cup F^{(i)}$; otherwise let $V_2 = V_2 \cup F^{(i)}$.
   (c) Let $D^{(i+1)} = \bar{F}^{(i)}$, and increment i by 1.
3. Return V_1 and V_2.

Proof of Lemma 3.4: Let t be the largest integer such that $D^{(t)}$ is not empty. To prove that the algorithm is correct, we will first show that at the beginning of the i-th loop in Step 2, for all $i \in \{0, \dots, t-1\}$, the following holds:

$$\min(|S \cap V_1|, |S \cap V_2|) + |S \cap F^{(i)}| \le |S|/2 .$$

Indeed, since $|S \cap F^{(i)}| \le |S \cap \bar{F}^{(i)}|$, we get that $\min(|S \cap V_1|, |S \cap V_2|) + |S \cap F^{(i)}|$ is at most

$$(|S \cap V_1| + |S \cap V_2| + |S \cap F^{(i)}| + |S \cap \bar{F}^{(i)}|)/2 \le |S|/2 .$$

By our procedure, $F^{(t)}$ contains a single vertex which will be assigned to the V_i such that $|S \cap V_i|$ is smaller. It follows that $|S \cap V_1|, |S \cap V_2| \le \lceil |S|/2 \rceil$ is an invariant maintained throughout the algorithm's iterations.

We will now prove that the separator size is $O(|G|^\alpha)$. First, note that $|D^{(i)}| \le \lceil |D^{(i-1)}|/2 \rceil \le (|D^{(i-1)}| + 1)/2$, hence $|D^{(i)}| < |G|/2^i + 1$. Thus, the separator size is

$$O(\sum_{i=0}^{t-1} |D^{(i)}|^\alpha) = O(\sum_{i=0}^{t-1} (|G|/2^i)^\alpha) = O(|G|^\alpha) .$$

□

Using this algorithm, we can prove the following lemma:

Lemma 3.5 *Let $G = (V, E)$ be a bounded degree graph in $\mathcal{G}(\alpha)$. Let S be a subset of V. Let ϵ satisfy $1 \geq \epsilon > 2/|G|$. Then, one can find an $O(|G|^\alpha/\epsilon^{1-\alpha})$-separator that divides G into two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|S \cap V_1|, |S \cap V_2| \leq |S|/2 + 1/\epsilon$, and $|V_1|, |V_2| \leq (1 + \epsilon)|V|/2$.*

Proof: Let T be an integer such that $2/(T+1) \leq \epsilon < 2/T$. Find a T–way partition $G'_1, \ldots, G'_T$ of G. By Lemma 3.1 this can be done so that the number of edges removed is $O(T^{1-\alpha}|G|^\alpha)$. Now, divide each $G'_i = (V'_i, E'_i)$ into two subgraphs $G'_{i,1} = (V'_{i,1}, E'_{i,1})$ and $G'_{i,2} = (V'_{i,2}, E'_{i,2})$ by Lemma 3.4, so that $|S \cap V'_{i,1}|, |S \cap V'_{i,2}| \in \{\lfloor |S \cap V'_i|/2 \rfloor, \lceil |S \cap V'_i|/2 \rceil\}$ and the cut–size is $O(|G_i|^\alpha) = O((|G|/T)^\alpha)$. The total number of edges removed in order to generate the $2T$ subgraphs $G'_{i,t}$ is $O(T^{1-\alpha}|G|^\alpha) = O(|G|^\alpha/\epsilon^{1-\alpha})$.

Without loss of generality, assume $|G'_{i,1}| \leq |G'_{i,2}|$. Consider the following procedure for dividing G into two subgraphs G_1 and G_2 satisfying the conditions stated in the lemma:

1. Let G_1, G_2 be empty graphs.

2. For $i = 1$ to T,

   If $|G_1| \geq |G_2|$, then let $G_1 = G_1 \cup G'_{i,1}$ and $G_2 = G_2 \cup G'_{i,2}$; otherwise let $G_1 = G_1 \cup G'_{i,2}$ and $G_2 = G_2 \cup G'_{i,1}$.

Since $||S \cap V'_{i,1}| - |S \cap V'_{i,2}|| \leq 1$, it follows that $||S \cap V_2| - |S \cap V_1|| \leq T$. Hence $|S \cap V_1|, |S \cap V_2| \leq |S|/2 + 1/\epsilon$. Moreover, there are at most $O(|G|^\alpha/\epsilon^{1-\alpha})$ edges of G connecting G_1 and G_2. To see that $|V_1|, |V_2| \leq (1+\epsilon)|G|/2$, observe that at the end of every iteration of the for–loop in the above procedure, $||G_1| - |G_2|| \leq \lceil |G|/T \rceil$.

□

3.2 An Algorithm for and the Proof of the Main Theorem

Let $G = (V, E)$ be a graph. Let ϵ be a constant satisfying the conditions of Lemma 3.5, $\psi \cdot \epsilon > 2k/|G|$, and $\Theta = \psi|G|/k$. Consider the following recursive procedure:

Algorithm: `min-max-boundary-partition`(G, Θ, ϵ)

1. If $|G| \leq \Theta$ then return G.
2. Apply the procedure of Lemma 3.5 to divide G into $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ where S is chosen to be the set of all boundary–vertices in G (at the first level of the recursion there are no boundary vertices, so we can just use an ordinary separator).
3. Let the set of boundary–vertices of G_1 and G_2 be those boundary–vertices inherited from G and those produced by the partition of the previous step.
4. Recursively call `min-max-boundary-partition`(G_1, Θ, ϵ).
5. Recursively call `min-max-boundary-partition`(G_2, Θ, ϵ).

If more than k subgraphs were generated, rename them $G_0, \ldots, G_{m-1}$, ordered by size, largest to smallest. Merge together subgraphs whose indices are equivalent modulo k.

We now prove our main separator theorem.

Proof of Theorem 3.2: Run the recursive procedure above with $\epsilon = \frac{1}{2}(1 - 2^{1-1/\alpha})$. This defines a separator tree T. The size of the subgraph at a leaf is at least $(1-\epsilon)\psi(|G|/k)/2$ and at most $\psi|G|/k$. The graph associated to the root of the separator tree is G itself. Let the *level* of a node in T be its distance from the root. Let c' be a constant such that every graph H in $\mathcal{G}(\alpha)$ has a separator of cut size at most $c'|H|^\alpha$. We now prove, by induction on the levels of the separator tree, that there is a constant c such that for every node v of T, $\partial_V(G_v) \leq c|G_v|^\alpha/\epsilon^{1-\alpha}$. The claim is true for the two children of the root, provided $c \geq c'$, since we can find a bisection of G of cut size at most $c'|G|^\alpha$. Assume that the claim is true for every internal node v at level $i-1$. Let u and w be the two children of v. The algorithm divides G_v into G_u and G_w. Since $2/\epsilon \leq \psi|G|/k$ and $(1-\epsilon)\psi(|G|/k)/2 \leq |G_v|$, we have that $1/\epsilon \leq (1-\epsilon)\psi|G|/k \leq 2|G_v|$. Let c_1 be the constant hidden in the O-notation of Lemma 3.5. Hence, if G' denotes either G_u or G_w, we have that

$$\begin{aligned}
\partial_V(G') &\leq \partial_V(G_v)/2 + 1/\epsilon + c_1|G_v|^\alpha/\epsilon^{1-\alpha} \\
&\leq (c/2)|G_v|^\alpha/\epsilon^{1-\alpha} + 2^\alpha|G_v|^\alpha/\epsilon^{1-\alpha} + c_1|G_v|^\alpha/\epsilon^{1-\alpha} \\
&= (c/2 + c_1 + 2^\alpha)|G_v|^\alpha/\epsilon^{1-\alpha} \\
&\leq (2^\alpha(c/2 + c_1 + 2^\alpha)/(1-\epsilon)^\alpha)\,|G'|^\alpha/\epsilon^{1-\alpha}\,.
\end{aligned}$$

The last inequality follows since Lemma 3.5 ensures that $|G'| \geq (1-\epsilon)|G_v|/2$. To conclude the inductive proof choose c such that $c \geq 2^\alpha(c/2 + c_1 + 2^\alpha)/(1-\epsilon)^\alpha$, i.e.,

$$c \cdot ((1-\epsilon)^\alpha/2^\alpha - 1/2) \;\geq\; c_1 + 2^\alpha\,.$$

This can be done as long as ϵ is bounded away from $1 - 2^{1-1/\alpha}$, as is the case by our choice of ϵ.

So far, we have shown that all of the subgraphs $G_0, \ldots, G_{m-1}$ produced by the `min-max-boundary-partition` procedure have $\partial_V(G_i) \leq c|G_i|^\alpha/\epsilon^{1-\alpha}$. It remains to show that the k graphs produced by the merging procedure have size at most $(1+\psi)|G|/k$ and boundary at most $O(\frac{1}{\psi^{1-\alpha}}(|G|/k)^\alpha)$. Let $G'_0, \ldots, G'_{k-1}$, denote these k graphs, with G'_i being the union of the graphs G_l such that l is equivalent to i modulo k. To see that for all $i < j$, $|G'_i| - |G'_j| \leq \psi|G|/k$, observe that for every graph G_l that is merged into G'_i but the largest, there is a larger graph, $G_{l-i-(k-j)}$ that is merged into G_j. Thus, G'_i can be no larger than G'_j than the size of the largest graph merged into G'_i, which necessarily has size at most $\psi|G|/k$. It follows that $G'_1, \ldots, G'_k$ is a $(1+\psi, k)$-partition of G.

We now argue that $\partial_E(G'_i) = O(\frac{1}{\psi^{1-\alpha}\epsilon^{1-\alpha}}(|G|/k)^\alpha)$. Since G is a bounded-degree graph it suffices to show that $\partial_V(G'_i) = O(\frac{1}{\psi^{1-\alpha}\epsilon^{1-\alpha}}(|G|/k)^\alpha)$. To see this, first observe that G'_i has size at most $(1+\psi)|G|/k$ and it is the union of subgraphs of size at least $(1-\epsilon)\psi(|G|/k)/2$.

Hence it is the union of at most $O(1/\psi)$ subgraphs of size at most $\psi|G|/k$. The number of boundary–vertices in each of these subgraphs is $O(\psi^{\alpha}(|G|/k)^{\alpha}/\epsilon^{1-\alpha})$. The desired conclusion follows. □

Corollary 3.6 *Let k be an integer such that $k > 1$. Let ψ be a constant such that $\psi \cdot (1 - 2^{1-1/\alpha}) > 4k/|G|$. Then, every n-node well–shaped mesh or nearest neighbor graph has a $(1+\psi, k)$-partition P with* $\text{max-boundary}_E(P) = O((n/k)^{1-1/d})$*; every n-node bounded–degree planar graph, graph with bounded genus, and graph with bounded forbidden minor has a $(1+\psi, k)$-partition P with* $\text{max-boundary}_E(P) = O(\sqrt{n/k})$.

4 Partitioning Weighted Graphs

In the following two situations, it is necessary to partition weighted graphs. In adaptive numerical formulation, in order to efficiently achieve a desired solution accuracy, sophisticated *adaptive* strategies that vary the solution or discretization technique within each finite element are used. For example, the p-refinement technique applies a higher order basis function in those elements having a rapidly changing solution or a large error. The h-refinement technique involves subdivision of the mesh elements themselves. (The $p-$ and hybrid hp-refinement [4] techniques can be used to efficiently find accurate solutions to problems in areas such as computational plasticity.) Strategies such as $p-$ and hp-refinement may cause the work to vary at different elements in the domain. This variation may be as high as one or two orders of magnitude [4].

In N-body simulations for non–uniformally distributed particles [2, 8, 19], particles will be grouped into clusters based on their geometric location. The interaction between particles in a pair of well–separated clusters will be approximated by the interaction between their clusters. The amount of calculation associated with some cluster/particle may be much higher than the amount of calculation needed in some other cluster/particle.

Consider a graph where every vertex is assigned a weight that is proportional to the amount of computation needed at the vertex. Let the *total weight* of a graph be the sum of the weight of its vertices. Rather than partitioning the graph into subgraphs of equal vertex size we would now like to partition it into subgraphs with "equal" total weight. However, partitioning according to weights alone may cause an imbalance in the size of the resulting subgraphs. In some applications, this may cause an imbalance in local memory requirements since, in general, all vertices need a similar amount of storage even though the computational work associated with them may vary. We consider the problem of partitioning vertex–weighted graphs into subgraphs with balanced weights and vertex-set sizes and minimal maximum boundary.

4.1 Simultaneous Partition of Vertices and Weights

Let $G = (V, E, w)$ be a vertex-weighted graph, where $w : V \to \mathbb{R}_+$ is a positive weight vector. For any subgraph $G' = (V', E')$ of G, we denote by $w(G')$ or $w(V')$ the total weight of G', i.e., $w(G') = w(V') = \sum_{v \in V'} w(v)$.

A variant of the following lemma was given in Lipton and Tarjan [12].

Lemma 4.1 *Let* $0 < \lambda < 1/2$. *Let* $G = (V, E)$ *be a bounded–degree graph in* $\mathcal{G}(\alpha)$ *and* $w : V \to \mathbb{R}_+$ *be a weight–vector such that* $w(v) < \lambda w(G)$ *for all* $v \in V$. *Then, one can find an* $O(|G|^\alpha)$*-separator that divides* G *into two subgraphs* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *such that* $w(G_1), w(G_2) \leq (1 + \lambda)w(G)/2$.

The following is an algorithm for constructing a partition with the properties stated in the lemma.

1. Let $D^{(0)} = G$, $V_1 = V_2 = \emptyset$, and $i = 0$.
2. Repeat until $D^{(i)}$ is an empty graph.
   (a) If $|D^{(i)}| = 1$, then let $F^{(i)} = D^{(i)}$ and $\bar{F}^{(i)} = \emptyset$; otherwise find a bisection of cut–size $O(|D^{(i)}|^\alpha)$ that divides the vertex set of $D^{(i)}$ into $F^{(i)}$ and $\bar{F}^{(i)}$ and assume, without loss of generality, that $w(F^{(i)}) \leq w(\bar{F}^{(i)})$.
   (b) If $w(V_1) \leq w(V_2)$, let $V_1 = V_1 \cup F^{(i)}$; otherwise let $V_2 = V_2 \cup F^{(i)}$.
   (c) Let $D^{(i+1)} = \bar{F}^{(i)}$ and increment i by 1.
3. Return V_1 and V_2.

The proof of the lemma is similar to that of Lemma 3.4.

Lemma 4.2 *Let* $G = (V, E)$ *be a bounded–degree graph in* $\mathcal{G}(\alpha)$. *Let* ϵ *satisfy* $1 \geq \epsilon > 2/|G|$ *and* $0 < \lambda < 1/2$. *Let* $w : V \to \mathbb{R}_+$ *be a weight–vector such that* $w(v) < \lambda w(G)$ *for all* $v \in V$. *Then, one can find an* $O(|G|^\alpha/\epsilon^{1-\alpha})$*-separator that divides* G *into two subgraphs* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *such that* $|V_1|, |V_2| \leq (1 + \epsilon)|V|/2$ *and* $w(G_1), w(G_2) \leq w(G)/2 + \lambda w(G)/\epsilon$.

The proof is similar to that of Lemma 3.5. The only difference is that we apply Lemma 4.1 to divide each G'_i instead of Lemma 3.4.

Proof: Let T be an integer such that $2/(T + 1) \leq \epsilon < 2/T$. Find a T–way partition $G'_1, \ldots, G'_T$ of G. By Lemma 3.1 this can be done so that the number of edges removed is $O(T^{1-\alpha}|G|^\alpha)$. Now, for $i \in \{1, \ldots, T\}$, find a $O(|G_i|^\alpha) = O((|G|/T)^\alpha)$ separator that divides $G'_i = (V'_i, E'_i)$ into two subgraphs $G'_{i,1}$ and $G'_{i,2}$ by Lemma 4.1, so that $w(G'_{i,1}), w(G'_{i,2}) \leq (1 + \lambda)w(G_i)/2$. Observe that the total number of edges removed in order to generate the $2T$ subgraphs $G'_{i,t}$ is $O(T^{1-\alpha}|G|^\alpha) = O(|G|^\alpha/\epsilon^{1-\alpha})$. Without loss of generality, assume $w(G'_{i,1}) \leq w(G'_{i,2})$. Consider the following procedure for dividing G into two subgraphs G_1 and G_2 satisfying the conditions stated in the lemma:

1. Let G_1, G_2 be empty graphs.
2. For $i = 1$ to T,

If $|G_1| \geq |G_2|$, then let $G_1 = G_1 \cup G'_{i,1}$ and $G_2 = G_2 \cup G'_{i,2}$; otherwise let $G_1 = G_1 \cup G'_{i,2}$ and $G_2 = G_2 \cup G'_{i,1}$.

Since $|w(G'_{i,1}) - w(G'_{i,2})| \leq \lambda w(G_i)$, it follows that $|w(G_2) - w(G_1)| \leq \lambda T w(G)$. Hence $w(G_1), w(G_2) \leq (1+\lambda T)w(G)/2 \leq w(G)/2 + \lambda w(G)/\epsilon$. Moreover, there are at most $O(|G|^\alpha/\epsilon^{1-\alpha})$ edges of G connecting G_1 and G_2.

To see that $|V_1|, |V_2| \leq (1+\epsilon)|G|/2$, observe, as in the proof of Lemma 3.5, that at the end of every iteration of the for–loop in the above procedure, $||G_1| - |G_2|| \leq \lceil |G|/T \rceil$. □

Let k be an integer such that $k > 1$. Let $G = (V, E, w)$ be a vertex–weighted graph. Let $P = \{G_1, \ldots, G_k\}$ be a collection of subgraphs $G_i = (V_i, E_i)$ of G that have disjoint vertex sets. We say that P is a (β, δ, k)-*partition* of G if the V_i's cover all of V, and for all $i \in \{1, \ldots, k\}$ it holds that $|G_i| \leq \beta \lceil |G|/k \rceil$ and $w(G_i) \leq \delta w(G)/k$.

Corollary 4.3 *Let $k \geq 16$ be a power of 4 such that $|G| \geq 13k^{9/4}$. Let $G = (V, E)$ be a bounded–degree graph in $\mathcal{G}(\alpha)$ and $w : V \to \mathbb{R}_+$ be a weight–vector such that $w(v) \leq (1/84)w(G)/k^{9/4}$ for all $v \in V$. Then, a $(3/2, 3/2, k)$-partition $P = \{G_1, \ldots, G_k\}$ of G can be constructed where* total-boundary$_E(P) = O(k^{2(1-\alpha)}|G|^\alpha)$ *and $|G_i| \geq |G|/2k$, for all $i \in \{1, \ldots, k\}$.*

Proof: Let ϵ be such that $(1+\epsilon)^4 = 4/3$ (i.e., $\epsilon \approx 0.0746$) and let $\lambda = 2\epsilon^2$. Also, let $t = 2\log k$ and $k' = \sqrt{2^t k} = k^{3/2}$. Observe that since k is a power of four, t, k', k'/k, and $2^t/k'$ are integers.

Recursively apply Lemma 4.2 until 2^t subgraphs $G''_i = (V''_i, E''_i)$, $i \in \{0, \ldots, 2^t - 1\}$, are generated. In order to perform the recursion we need that $\epsilon \cdot |G|(\frac{1-\epsilon}{2})^{t-1} > 2$ and $\lambda \cdot w(G)(\frac{1-2\lambda/\epsilon}{2})^{t-1} > w(v)$ for all $v \in V$. Both inequalities are guaranteed by the hypothesis. Note that the total number of edges removed throughout the recursion is $O(\sum_{i=0}^{t-1} 2^i(|G|/2^i)^\alpha) = O(|G|^\alpha 2^{t(1-\alpha)}) = O(|G|^\alpha k^{2(1-\alpha)})$.

Let $\widetilde{\epsilon} = (1+\epsilon)/2$ and $\widetilde{\lambda} = (1+2\lambda/\epsilon)/2$. Thus, $|V''_i| \leq \widetilde{\epsilon}^t|V|$ and $w(G''_i) \leq \widetilde{\lambda}^t w(G)$. Hence, $||V''_i| - |V''_j|| \leq \widetilde{\epsilon}^t|V|$ and $|w(G''_i) - w(G''_j)| \leq \widetilde{\lambda}^t w(G)$. Rename the subgraphs $G''_0, \ldots, G''_{2^t-1}$ according to size. Merge together subgraphs whose indices are equivalent modulo k'. Let $G'_i = (V'_i, E'_i)$, $i \in \{0, \ldots, k'-1\}$, be the graphs generated in this way. Note that $||V'_i| - |V'_j|| \leq \widetilde{\epsilon}^t|V|$ and $|w(G'_i) - w(G'_j)| \leq (2^t/k')\widetilde{\lambda}^t w(G)$. Now, rename the subgraphs $G_0, \ldots, G_{k'-1}$ ordered by weight and merge together those whose indices are equivalent modulo k. Let $G_i = (V_i, E_i)$, $i \in \{0, \ldots, k-1\}$, be the graphs generated in this way. Note that $||V_i| - |V_j|| \leq (k'/k)\widetilde{\epsilon}^t|V|$ and $|w(G_i) - w(G_j)| \leq (2^t/k')\widetilde{\lambda}^t w(G)$. It follows that $(1 - k(k'/k)\widetilde{\epsilon}^t)|V|/k \leq |V_i| \leq (1 + k(k'/k)\widetilde{\epsilon}^t)|V|/k$, and $w(G_i) \leq (1 + k(2^t/k')\widetilde{\lambda}^t)w(G)/k$. But, $k(k'/k)\widetilde{\epsilon}^t = ((1+\epsilon)^4/2)^{t/4} \leq (1/2)^{t/8} \leq 1/2$. Analogously, $k(2^t/k')\widetilde{\lambda}^t \leq 1/2$. Hence, $|V|/(2k) \leq |V_i| \leq 3|V|/(2k)$ and $w(G_i) \leq 3w(G)/(2k)$. □

4.2 Min–Max–Boundary Partition of Weighted Graphs

Theorem 4.4 *Let k be an integer such that $k > 1$. Let $G = (V, E)$ be a bounded–degree graph in $\mathcal{G}(\alpha)$. Let $w : V \to \mathbb{R}_+$ be a weight–vector such that* $\frac{w(G)}{w(v)} = \Omega\left(\frac{k}{\psi^2}\left(\log(\frac{k}{\psi^2})\right)^{9/4}\right)$

for all $v \in V$*. Let* $|G| = \Omega\left(\frac{k}{\psi^2}\left(\log(\frac{k}{\psi^2})\right)^{9/4}\right)$*. Then, a* $(1+\psi, 1+\psi, k)$*-partition* P *of* G *such that* $\text{max-boundary}_E(P) = O\left(\left(\frac{1}{\psi}\log(\frac{k}{\psi^2})\right)^{2(1-\alpha)}\left(\frac{|G|}{k}\right)^{\alpha}\right)$ *can be constructed.*

To prove this theorem we follow an argument analogous to the one used in Section 3.2 to prove Theorem 3.2. The algorithm recursively applies the following lemma to simultaneously partition weights, vertices, and boundary.

Lemma 4.5 *Let* $G = (V, E)$ *be a bounded–degree graph in* $\mathcal{G}(\alpha)$*. Let* $w : V \to \mathbb{R}_+$ *be a weight–vector such that* $w(v) < \lambda w(G)$ *for all* $v \in V$*. Let* $S \subseteq V$ *be a subset of* V*. Let* ϵ *be such that* $(\max\{13/|G|, 84\lambda\})^{4/9}/2 < \epsilon \leq 1/128$*. Then, one can find an* $O(|G|^{\alpha}/\epsilon^{2(1-\alpha)})$*-separator that divides* G *into two subgraphs* $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *such that* $|S \cap V_1|, |S \cap V_2| \leq |S|/2 + 1/2\epsilon^2$*,* $|V_1|, |V_2| \leq (1+\epsilon)|V|/2$*, and* $w(G_1), w(G_2) \leq (1+12\epsilon)w(G)/2$*.*

Proof: The proof follows the basic idea developed in the proof of Lemma 3.5. So we only highlight the difference. Let T be a power of 4 such that $1/(8\epsilon) \leq T \leq 1/(2\epsilon)$. Find a T–way partition $G'_1, \ldots, G'_T$ of G by Corollary 4.3. This can be done so the total number of edges removed is $O(T^{2(1-\alpha)}|G|^{\alpha}) = O(|G|^{\alpha}/\epsilon^{2(1-\alpha)})$. Recall that $|G|/(2T) \leq |G'_i| \leq (3/2)\lceil |G|/T \rceil$ and $w(G'_i) \leq (3/2)w(G)/T$, for all $i \in \{1, \ldots, T\}$. We need to verify that Corollary 4.3 is indeed applicable. In other words we need to show that $T \geq 16$, $|G| \geq 13T^{9/4}$, and $w(v) \leq (1/84)w(G)/T^{9/4}$ for all $v \in V$. Indeed, since $\epsilon \leq 1/128$ and $1/(8\epsilon) \leq T$, we get that $T \geq 16$. Since $(13/|G|)^{4/9}/2 \leq \epsilon$ and $T \leq 1/(2\epsilon)$, we get that $13T^{9/4} \leq |G|$. Finally, since $(84\lambda)^{4/9}/2 \leq \epsilon$ and $T \leq 1/(2\epsilon)$, we get that $w(v) \leq (1/84)w(G)/T^{9/4}$ for all $v \in V$.

We now show that we can divide every $G'_1, \ldots, G'_T$ into two subgraphs using Lemma 3.5. We need to show that $\epsilon > 2/|G'_i|$. Indeed, since $(2/|G|)^{1/2} \leq (13/|G|)^{4/9}/2 < \epsilon$, we have $|G'_i| \geq |G|/(2T) \geq |G|\epsilon > 2/\epsilon$. Hence, we can divide each $G'_i = (V'_i, E'_i)$ into two subgraphs $G'_{i,1} = (V'_{i,1}, E'_{i,1})$ and $G'_{i,2} = (V'_{i,2}, E'_{i,2})$ by Lemma 3.5 so that $|S \cap V'_{i,1}|, |S \cap V'_{i,2}| \leq |S \cap V'_i|/2 + 1/\epsilon$, and $|V'_{i,1}|, |V'_{i,2}| \leq (1+\epsilon)|V'_i|/2$. The T applications of Lemma 3.5 can be done so the total number of edges removed is $O(\sum_{i=1}^{T}|G'_i|^{\alpha}/\epsilon^{1-\alpha}) = O(T^{1-\alpha}|G|^{\alpha}/\epsilon^{1-\alpha}) = O(|G|^{\alpha}/\epsilon^{2(1-\alpha)})$.

Without loss of generality, assume $w(G'_{i,1}) \leq w(G'_{i,2})$. Consider the following procedure for dividing G into two subgraphs G_1 and G_2 satisfying the conditions stated in the lemma:

1. Let G_1, G_2 be empty graphs.
2. For $i = 1$ to T,

   If $w(G_1) \geq w(G_2)$, then let $G_1 = G_1 \cup G'_{i,1}$ and $G_2 = G_2 \cup G'_{i,2}$; otherwise let $G_1 = G_1 \cup G'_{i,2}$ and $G_2 = G_2 \cup G'_{i,1}$.

Since $||S \cap V'_{i,1}| - |S \cap V'_{i,2}|| \leq 2/\epsilon$, it follows that $||S \cap V_2| - |S \cap V_1|| \leq 2T/\epsilon$. Hence $|S \cap V_1|, |S \cap V_2| \leq |S|/2 + T/\epsilon \leq |S|/2 + 1/(2\epsilon^2)$. Moreover, there are at most $O(T^{1-\alpha}|G|^{\alpha}/\epsilon^{1-\alpha}) = O(|G|^{\alpha}/\epsilon^{2(1-\alpha)})$ edges of G connecting G_1 and G_2. In addition, because $||V'_{i,1}| - |V'_{i,2}|| \leq \epsilon|V'_i|$, it follows that $||V_1| - |V_2|| \leq \epsilon\sum_{i=1}^{T}|V'_i| = \epsilon|V|$. Hence $|V_1|, |V_2| \leq (1+\epsilon)|V|/2$. Since $|w(G'_{i,1}) - w(G'_{i,2})| \leq w(G'_i) \leq (3/2)w(G)/T$, by a similar argument as the one given in Lemma 3.5 we can show that $|w(G_1) - w(G_2)| \leq (3/2)w(G)/T$. Hence, $w(G_1), w(G_2) \leq (1 + \frac{1}{T}(3/2))w(G)/2 \leq (1+12\epsilon)w(G)/2$. □

5 Conclusions

EXPERIMENTAL RESULTS: To assess the quality of our algorithms, we conducted experiments on several sample meshes, using the Geometric Mesh Partitioning Toolbox developed by Gilbert and Teng [10].

Mesh	Description	Mesh Type	Grading	Vertices	Edges
AIRFOIL2	Three-element airfoil	2-D triangles	1.3×10^5	4720	13722
TRIANGLE	Equilateral triangle	2-D triangles, all same size	1.0×10^0	5050	14850
AIRFOIL3	Four-element airfoil	2-D triangles	3.0×10^4	15606	45878
PWT	Pressurized wind tunnel	Thin shell in 3-space	1.3×10^2	36519	144794
BODY	Automobile body	3-D volumes and surfaces	9.5×10^2	45087	163734
WAVE	Space around airplane	3-D volumes and surfaces	3.9×10^5	156317	1059331

Table 1: Test problems. "Grading" is the ratio of longest to shortest edge lengths.

Table 1 lists the meshes. AIRFOIL2 and AIRFOIL3 are highly graded meshes of well-shaped 2D triangles around cross sections of airfoils, from Barth and Jesperson [3]. TRIANGLE is a 2D mesh of equilateral triangles, all the same size, generated by `gridt` in Matlab. The value of α for these meshes is 1/2. PWT is a mesh of 3D elements that discretize a thin shell. BODY is another 3D mesh with some "thin shell" parts. We obtained these two meshes from Horst Simon at NASA. Because of the 2D embedding in 3D, the mesh partitioning algorithm in general generates partitions with $\alpha = (1.5/2.5)$. WAVE is a highly graded mesh that fills the space around an object in 3D, which we obtained from Steve Hammond at NCAR. In this case, the value of α is 2/3.

Mesh	2-way	16-way	128-way	e^α	α
AIRFOIL2	100	31	15	117	1/2
TRIANGLE	144	55	19	122	1/2
AIRFOIL3	152	61	20	214	1/2
PWT	529	151	55	1248	1.5/2.5
BODY	834	265	75	1344	1.5/2.5
WAVE	10377	3013	721	10391	2/3

Table 2: Maximum boundary size for multi-way partitions.

On these samples of finite element meshes in both two and three dimensions, the experiments show that the boundary size is bounded from above by $1.5(|E|/k)^\alpha$.

EXTENSIONS: An alternative way to partition a graph is by removing vertices rather than by removing edges. Vertex–based decomposition has been used in nested dissection for solving sparse linear systems [13] and overlapping domain decomposition [9]. This motivates the following vertex-based decomposition problem. Given a graph $G = (V, E)$ and an integer $k > 1$, we say that $D = \{V_1, \ldots, V_k\}$ is a (β, k)-*decomposition* of V if the subgraphs $G_i = (V_i, E_i)$

of G induced by the V_i's are such that $\cup_{i=1}^k V_i = V$, $\cup_{i=1}^k E_i = E$, and $|V_i| \leq \beta\lceil |V|/k \rceil$, for all $i \in \{1, \ldots, k\}$. Note that in such a decomposition $G_1, \ldots, G_k$ may be pair–wise overlapping. Let $\partial(G_i)$ denote the set of vertices in V_i that are also nodes of some other subgraph G_j, $j \neq i$. As in multi–way graph partitioning, we consider the following two costs associated with a (β, k)-decomposition:

$$\begin{aligned} \text{total-boundary}_V(D) &= \sum_{i=1}^{k} |\partial(G_i)| \\ \text{max-boundary}_V(D) &= \max_{i=1,\ldots,k} |\partial(G_i)| \,. \end{aligned}$$

Extensions of the arguments presented in Section 3 yield vertex–separator results similar in spirit to those stated in Lemma 3.5 and Theorem 3.2.

References

[1] N. Alon, P. Seymour, and R. Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the 22th Annual ACM Symposium on Theory of Computing*, pages 293–299. ACM, 1990.

[2] J. Barnes and P. Hut. A hierarchical $O(n \log n)$ force calculation algorithm. *Nature*, (324):446–449, 1986.

[3] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting*. AIAA, 1989.

[4] M. Benantar, R. Biswas, J. E. Flaherty, and M. S. Shephard. Parallel computation with adaptive methods for elliptic and hyperbolic systems. *Comp. Methods Applied Mech. and Eng.*, pages 73–93, 1990.

[5] M. Bern, D. Eppstein, and J. R. Gilbert. Provably good mesh generation. In *31-st Annual Symposium on Foundations of Computer Science*, pages 231–241. IEEE, 1990.

[6] S. N. Bhatt and F. T. Leighton. A framework for solving VLSI graph layout Problems. *J. Comp. and System Sciences*, 28, pages 300-343, 1984.

[7] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. An iterative method for elliptic problems on regions partitioned into substructures. *Math. Comp.*, 46:361–9, 1986.

[8] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Statist. Comput.* 9:669–686, 1988.

[9] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. *Acta Numerica*, pages 61–144, 1994.

[10] J. R. Gilbert, G. L. Miller, and S.-H. Teng. Geometric mesh partitioning: Implementation and experiments. In *SIAM J. Sci. Comput.*, to appear, 1999.

[11] J.R. Gilbert, J.P. Hutchinson, and R.E. Tarjan. A separation theorem for graphs of bounded genus. *Journal of Algorithms*, 5:391–407, 1984.

[12] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. of Appl. Math.*, 36:177–189, April 1979.

[13] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM J. on Numerical Analysis*, 16:346–358, 1979.

[14] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Finite element meshes and geometric separators. *SIAM J. Sci. Comput.*, 19(2): 364–386, 1998.

[15] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighborhood graphs. *J. ACM*, 44(1): 1–29, January. 1997.

[16] H. D. Simon and S.-H. Teng. How good is recursive bisection? *SIAM J. Sci. Comput.*, 18(5): 1436–1445, September 1997.

[17] D. A. Spielman and S.-H. Teng. Spectral partitioning works: planar graphs and finite element meshes. In *Proceedings of the 37-th Annual Symposium on Foundation of Computer Science*, pages 96–107, IEEE, 1996,

[18] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[19] S.-H. Teng. Provably good partitioning and load balancing algorithms for parallel adaptive n-body simulation. *SIAM J. Scientific Computing*, 19 (2), 635–656, March, 1998.