Geometric Embeddings for Faster and Better Multi-Way Netlist Partitioning*

C. J. Alpert and A. B. Kahng

Computer Science Department, University of California at Los Angeles, Los Angeles, CA 90024-1596

Abstract

We give new, effective algorithms for k-way circuit partitioning in the two regimes of $k \ll n$ and $k = \Theta(n)$, where n is the number of modules in the circuit. We show that partitioning an appropriately designed geometric embedding *of the netlist, rather than a traditional graph representation, yields improved results as well as large speedups. We derive d-dimensional geometric embeddings of the netlist via (i) a new "partitioning-specific" net model for constructing the Laplacian of the netlist, and (ii) computation of d eigenvectors of the netlist Laplacian; we then apply (iii) fast top-down and bottom-up geometric clustering methods.*

1 Preliminaries

In top-down layout synthesis of complex VLSI systems, the goal of partitioning/clustering is to reveal the *natural* circuit structure, via a decomposition into k subcircuits which minimizes connectivity between subcircuits. A generic problem statement is as follows:

k-Way Partitioning: Given a circuit netlist $G = (V, E)$ with $|V| = n$ modules, construct a *k-way partitioning* P_k which divides the modules into a set of k disjoint subcircuits $\{C_1, C_2, \ldots, C_k\}$, such that a given objective function $F(P_k)$ is minimized.

The relative sizes of k and n determine one of two distinct multi-way partitioning applications: the *small-k partitioning* (SKP) problem or the *large-k partitioning* (LKP) problem.

1.1 The SKP Problem

In the SKP regime, $k \ll n$ and is in practice bounded by a small constant, e.g., $k \leq 10$. The SKP problem arises in high-level system partitioning and floorplanning. Early approaches involved seeded epitaxial growth, extensions of the Fiduccia-Mattheyses iterative interchange bipartitioning algorithm [18], and a primal-dual iteration motivated by a generalization of the minimum ratio cut metric [21]. These methods use simple objective functions based only on the number of nets crossing partition boundaries; they moreover require the partition sizes to be specified in advance. Recently, Yeh et al. [22] proposed a "shortest-path clustering" (SPC) method, where "shortest paths" between random pairs of modules are iteratively deleted from the netlist graph until it becomes disconnected into multiple components (i.e., the clusters). This algorithm probabilistically captures the relationship between multicommodity flow and the minimum ratio cut. SPC solutions are of high quality when measured by the *cluster ratio*, which Yeh et al. argue to be the "proper" k-way generalization of the ratio cut objective.

Minimum Cluster Ratio SKP: Find a clustering $P_k = \{C_1, C_2, \ldots, C_k\}$, $2 \leq k \leq |V|$, that minimizes

$$F(P_k) = \frac{c(P_k)}{\sum_{j=i+1}^{k} \sum_{i=1}^{k-1} |C_i| \times |C_j|}.$$

Here $c(P_k)$ is the number of nets which cross between two or more of the clusters of P_k. The SPC method, while of high quality, requires shortest-path computations in the netlist graph and exhaustive enumeration of all partitions of disconnected components obtained through the shortest-path deletion; its $O(mn \log n)$ time complexity also depends on two "accuracy" parameters b and $\frac{1}{\Delta}$.

In [4], Chan et al. generalize the result of [10] from 2-way to k-way ratio cut partitioning. Chan et al. use the first k eigenvectors of the netlist Laplacian to construct an orthogonal "projector" which maps an n-dimensional space into a k-dimensional space. Ideally, n elementary unit vectors in the n-space (modules) will be mapped to exactly k distinct points in the k-space (partitions) by this projector. Since this is not the case in practice, the authors of [4] use heuristic clustering methods in k-space to obtain a k-way partitioning. The approach requires additional matrix manipulations and a more complicated, netlist-based clustering methodology than our methods below.

The authors of [4] also generalize the ratio cut objective using a dimensionless *scaled cost*:

Minimum Scaled Cost SKP: Find a clustering $P_k = \{C_1, C_2, \ldots, C_k\}$, $2 \leq k \leq |V|$, that minimizes

$$F(P_k) = \frac{1}{n(k-1)} \sum_{i=1}^{k} \frac{E_i}{|C_i|}.$$

*Partial support for this work was provided by a Department of Defense Graduate Fellowship, by NSF MIP-9110696, NSF Young Investigator Award MIP-9257982, ARO DAAK-70-92-K-0001, and ARO DAAL-03-92-G-0050, and by Cadence Design Systems and Zycad Corporation under the State of California MICRO program.

Here, E_i is the number of signal nets crossing the boundary of the C_i partition.

We see that [4] and [22] each establish a generalization of the ratio cut concept. Each metric is robust, and automatically accounts for both cut nets and size balance among the partitions. Below, we compare our new methods against the results in [4] and [22], using their respective metrics.

1.2 The LKP Problem

In the LKP regime, $k = \Theta(n)$, e.g., $k = \frac{n}{5}$. The LKP problem arises with two-phase enhancements of Fiduccia-Mattheyses (F-M) [2] [11] [15]; a partitioning into small clusters induces a "compacted" netlist and reduces the solution space so that it can be searched more effectively. Thus, we wish to achieve LKP solutions with k just small enough for standard iterative approaches to once again become effective.

With this goal in mind, Bui et al. [2] proposed the "matching-based compaction" (MBC) algorithm, where the edges of a maximal random matching in the netlist graph induce a compacted instance of $\frac{n}{2}$ vertices, and the compaction is iterated until the problem size becomes manageable. After compaction, heuristic F-M bipartitioning is performed, and the netlist is re-expanded into a "flat" initial configuration for a second F-M phase.

Recently, Hagen et al. [11] developed the probabilistic RW-ST method which finds "natural clusters" via a self-tuning *random walk* in the circuit netlist. Strongly connected regions of the netlist are detected by multiple revisitations of modules within the walk. [11] reports that the RW-ST clusters lead to significant improvements in the performance of two-phase F-M, as compared to the MBC strategy. However, $\Theta(n^3)$ time is required to process a random walk of the recommended $\Theta(n^2)$ length.

1.3 Outline of Paper

The remainder of this paper develops our new methodology, which applies fast *geometric clustering* algorithms to a geometric embedding of the netlist. Phase I of our methodology constructs the embedding, i.e., we map each module of the netlist to a point in d-dimensional Euclidean space. The success of this procedure is strongly influenced by a new "partitioning-specific" net model used in constructing the weighted netlist graph. Phase II of our methodology computes a fast (top-down or bottom-up) clustering of the points in the embedding and then maps the clustering back to the original netlist, returning it as a k-way netlist partitioning. We propose use of a *min-diameter* clustering objective in conjunction with two fast methods, KC and AGG; these afford consistently good k-way netlist partitions in $O(n \log k)$ and $O(n^2)$ time respectively. Section 4 presents experimental data showing that our multi-way partitioning approach is quite robust. For SKP, we average 11.2% improvement over the work of Chan et al. in [4], and are also competitive with the method of [22] while maintaining better algorithmic complexity.[1] For LKP in the context of two-phase F-M bisection, our results significantly improve over the results of [2] [11], and achieve an overall 26.9% improvement over standard F-M bipartitioning. We conclude in Section 5 with directions for future research.

2 Phase I: Fast Embeddings

Many graph optimizations achieve speedups when the input is embedded in a geometric space (e.g., computing a minimum spanning tree (MST) [17]). For typical netlist sizes, speedups for the partitioning problem can become very significant. An embedding should be "distance-preserving": the distance between two points in the embedding should reflect the strength of connectivity between the corresponding modules. Following ideas of Hall [12], our approach embeds the netlist into d-dimensional Euclidean space via the well-established relationship between eigenvectors of the netlist Laplacian[2] and minimum ratio cut partitionings (or minimum squared wirelength placements) (see [11] [9] for surveys).

Every eigenvector of the Laplacian gives a *distinct*, one-dimensional spatial embedding of the circuit graph wherein strongly connected modules will tend to be placed close to each other. Because the squared wirelength of each eigenvector placement is given by its corresponding eigenvalue, the lowest eigenvalues will correspond to eigenvectors that are the most "distance-preserving". Thus, we derive a d-dimensional embedding of the netlist from the eigenvectors corresponding to the lowest d nonzero eigenvalues of the Laplacian. The i^{th} components of these d eigenvectors give the d coordinates $\Re^d$ of the i^{th} module. We compute embeddings using the same Lanczos code as [10]; Lanczos solves the sparse symmetric eigenproblem with $O(n^{1.4})$ runtime and is very competitive with the complexity of iterative partitioning methods. Moreover, our studies indicate that achieving a d-dimensional embedding (d eigenvectors) requires 1.15, 1.83, 2.00, 2.58, 2.64, 3.25, 3.29, 4.31 and 4.44 times the CPU cost of achieving a 1-dimensional embedding (i.e., just the first nontrivial eigenvector), for $d = 2, 3, \ldots, 10$ respectively.

A "Partitioning-Specific" Net Model

The success of our approach relies on an enhancement to the traditional eigenvector embedding, namely, a new clique net model for constructing the adjacency matrix A. Recall that the clique net model

[1] Recently, the authors of [4] have improved their algorithm by combining geometric and graph clustering [23]; see Section 5 for further discussion.

[2] We represent the circuit netlist by a simple undirected graph $G = (V, E)$ with $|V| = n$ vertices $v_1, \ldots, v_n$ representing the n modules, and edges in E capturing superposed hyperedges of the netlist hypergraph via a clique net model [15]. The $n \times n$ *adjacency matrix* $A = A(G)$ has A_{ij} equal to the weight of $\{v_i, v_j\} \in E$, and by convention $A_{ii} = 0$ for all $i = 1, \ldots, n$. If we let $d(v_i)$ denote the degree of node v_i (i.e., the sum of the weights of all edges incident to v_i), we obtain the $n \times n$ diagonal *degree matrix* D defined by $D_{ii} = d(v_i)$. The *Laplacian* of the netlist graph is given by $Q = D - A$.

represents a p-pin signal by a clique of $C(p,2)$ edges among its p modules; the cliques for all signal nets are superposed to yield the matrix A. Previous work has adopted a "standard" clique model [15], wherein a p-pin net contributes weight $\frac{1}{p-1}$ to each of $C(p,2)$ A_{ij} values.

This standard clique model seems to be motivated by the following consideration: in bipartitioning, when one p-pin signal net crosses the partition boundary, at least $(p-1)$ edges in its clique must be cut, hence the weight of cut edges will be $\geq (p-1) \cdot \frac{1}{(p-1)} = 1$. In other words, the standard net model ensures that a cut net will make a *minimum contribution* of 1 to the partitioning cost function. Unfortunately, a cut net can also contribute up to $\frac{p^2}{4} \cdot \frac{1}{(p-1)}$ to the partitioning objective, meaning that large nets will receive disproportionate attention in the partitioning process. To remedy this, we propose a new *partitioning-specific* clique net model, whereby any cut net will make an *expected contribution* of 1 to the cost function. Thus, all nets will receive uniform priority in the optimization. By enumerating all possible bipartitions of a net, it is simple to show that a uniform edge weight of $\frac{4}{p(p-1)}$ achieves this goal. We use this revised clique model in constructing our netlist Laplacian.[3]

3 Phase II: Geometric Clustering

The geometric embedding in Phase I maps the n netlist modules to n points in d-dimensional Euclidean space, denoted by $V = \{v_1, v_2, \ldots, v_n\} \subset \Re^d$. The goal of Phase II is to divide V into k disjoint *clusters*, which will correspond to the output partitions. We define a k-way clustering of V as $P_k = \{C_1, C_2, \ldots, C_k\}$ where $\cup_i C_i = V$ and the C_i are disjoint. A large body of work has established various objectives $f(P_k)$ that capture whether a given clustering of V is natural. Intuitively, clusters should be compact and well-separated. Clusters with small *diameter*, defined as $diam(C) = \max_{v_i, v_j \in C} d(v_i, v_j)$ (where $d(v_i, v_j)$ denotes the Euclidean distance between $v_i, v_j \in V$), satisfy the former goal; clusters with large *split*, defined as $split(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2} d(v_i, v_j)$, achieve the latter. Thus, standard objective functions in the literature include:

1. *Max-Split Clustering:* Maximize
$$f(P_k) = \min_{C_i, C_j \in P_k} \{split(C_i, C_j)\}$$

2. *Min-Diameter Clustering:* Minimize
$$f(P_k) = \max_{C \in P_k} \{diam(C)\}$$

3. *Min-Sum-Diameters Clustering:* Minimize
$$f(P_k) = \sum_{1=i}^{k} diam(C_i)$$

Each of these is an intuitively reasonable objective for our purposes. Previous works (e.g., [4] [12]) use "geometric clustering", but do not specify f. However, we have found the choice of f to be critical.

Formulation 1 is solved optimally by the Single-Linkage algorithm [14], which iteratively merges the closest pair of clusters. The same solution may be derived by removing the $k-1$ largest edges from the MST; the k connected components which remain form the clustering. The complexity of this approach is dominated by the MST construction, e.g., $O(n \log n)$ in the plane [17]. With respect to Formulations 2 and 3, the following results are known.

Fact 1: Formulations 2 and 3 are NP-Complete for $k \geq 3$ and $d \geq 2$ [16].

Fact 2: Solving Formulation 2 within a factor < 2 of optimal is NP-complete for $d \geq 3$ [5].

Fact 3: In general graphs whose edge weights do not satisfy the triangle inequality, neither Formulation 2 nor Formulation 3 may be approximated within any fixed constant factor of optimal for $k \geq 3$ [7].

When distances satisfy the triangle inequality, as they do in geometry, several heuristics achieve performance ratio 2 for Formulation 2. We use the simple k-center (KC) technique of Gonzalez [7], which runs in $O(nk)$ time and achieves this ratio:[4]

KC Algorithm
1. Initialize W, a set of cluster *centers*, to empty.
2. Choose some random v from V and add it to W.
3. While $\|W\| \leq k$, find $v \in V$ s.t. $\min_{w \in W} d(v, w)$ is maximized, and add it to W.
4. Form clusters $C_1, C_2, \ldots, C_k$ each containing a single point of W; place each $v \in V$ into the cluster of the closest $w_i \in W$.

Fact 2 states that Formulation 2 is NP-complete only for $k \geq 3$. Indeed, for $k = 2$, there are well-known efficient algorithms, e.g., bicoloring a *maximum spanning tree* divides the point set into an optimal min-diameter bipartitioning [8]. We may obtain a heuristic k-way partitioning by iteratively applying optimal bipartitioning to the largest current cluster; we call this the Divisive Min-Diameter approach. This technique illustrates the concept of a *hierarchical clustering* where successive clusterings P_i and P_{i+1} differ only in that a single $C \in P_i$ has been separated into $C_u, C_v \in P_{i+1}$.

[3] We ignore the $(0,p)$ and $(p,0)$ bipartitioning cases, since these do not actually cut the signal net. The actual uniform edge weight that we use is $\frac{4}{p(p-1)} \cdot \frac{2^p-2}{2^p}$. (Recent work in our group has more generally demonstrated the effects of net modeling on partition/placement quality.)

[4] Feder and Greene [5] used the KC construction as the basis for a more complicated but faster scheme which for $V \subset \Re^d$ yields a 2-approximate clustering in optimal $O(n \log k)$ time. However, we have chosen to use the $O(nk)$ KC algorithm for its simplicity.

Algorithm	Formulation	Ref	Number of Clusters (k) 9	8	7	6	5	4	3	2
RSBipart	–	[4]	48.8	44.5	40.6	37.1	41.0	36.6	27.3	14.6
KP (Chan et al.)	–	[4]	45.0	51.1	32.1	36.8	25.9	25.7	15.9	13.5
AGG	2	[14]	33.1	31.7	29.8	24.8	26.0	17.4	17.9	13.5
KC	2	[8]	34.6	33.6	34.4	30.7	27.5	16.4	17.4	13.5
Single-Linkage	1	[14]				49.5	39.2	39.9	23.7	13.5
Divisive Min-Diameter	2	[9]				59.5	46.8	42.9	32.8	13.5
Divisive Sum-Diameters	3	[13]				96.0	74.9	59.8	72.3	13.5
Agglom Sum-Diameters	3	–				154.5	127.2	88.3	72.3	13.5

Table 1: Comparison of the various clustering objectives and algorithms for the Primary1 benchmark netlist, using the partitioning-specific net model in constructing the Laplacian. Numbers reported give the Scaled Cost metric of Chan et al. [4].

While the top-down hierarchical approach reflects current practice for k-way nelist partitioning, it is also reasonable to construct a bottom-up hierarchical clustering. To this end, we propose to use an "agglomerative" algorithm AGG (cf. the Complete Linkage algorithm of Johnson [14]), which begins with each point in its own cluster and then iteratively merges a pair of clusters so as to minimize the increase in maximum cluster diameter. Benzecri [1] has given an $O(n^2)$ implementation of this algorithm based on constructions of chains of nearest neighbors.

AGG Algorithm

1. Initialize $P_n = \{C_1, C_2, \ldots, C_n\}$ s.t. each C_i contains exactly one vertex in V. Set $m = n$.
2. Given P_m, find clusters C_i, C_j s.t. $diam(C_i \cup C_j)$ is minimum. $P_{m-1} = (P_m \cup C) - C_i - C_j$.
3. Decrement m; if $m > k$ go to Step 2.

Formulation 3 may also be addressed by divisive and agglomerative approaches. [13] gives an efficient algorithm for Formulation 3 when $k = 2$; we apply this algorithm iteratively and call it Divisive Sum-Diameters. The Formulation 3 analogue of AGG is called Agglom Sum-Diameters. Table 1 summarizes the six algorithms that we have discussed in this section; the table also provides comparison data which we discuss next.

4 Results and Discussion

4.1 Results for the SKP Problem

Our experiments show clear differences in the partition quality afforded by the various objectives.

Table 1 shows the best results obtained by each of the six algorithms for the Primary1 benchmark. For $2 \leq k \leq 6$, the six algorithms are compared by taking the best clustering result over dimensions $1 \leq d \leq 5$, expressed using the Scaled Cost objective [4]. The Sum-Diameters objective tends to force our algorithms to isolate outliers in the embedding, thus generating poorly balanced clusters and high Scaled Cost. Single-Linkage slightly outperforms Divisive Min-Diameter, but both are noticeably inferior to AGG and KC. Since Single-Linkage solves Formulation 1 optimally, we conclude that minimization of split is not a good objective vis-a-vis the eigenvector-induced embeddings. We hypothesize that Divisive Min-Diameter's poor performance is due to the nature of our embeddings (dense with very few outliers), a problem class for which Divisive Min-Diameter often behaves poorly.

We also compare our results to the KP method of Chan et al. [4] for the Primary1 benchmark (we center on the Primary1 benchmark in the SKP discussion, since it is the only benchmark reported in [4]). Here AGG and KC represent the best values used over $1 \leq d \leq 10$. These values represent an average 11.2% win for KC, and an average 15.6% win for AGG, over KP. As would be expected, even larger improvements are achieved over the recursive spectral bipartitioning (RSBipart) results, which we quote from [4].

We digress momentarily to show the effect of the partitioning-specific net model in obtaining high-quality geometric embeddings that are suited to our clustering algorithms (Table 2). For AGG and KC, the partitioning-specific net model led to clearly improved results as measured by the Scaled Cost objective: for AGG, an average of 12.1% improvement in partitioning quality was observed, and KC results improved by an average of 17.4% with the new net model.

k	AGG $1/(p-1)$	AGG $4/p(p-1)$	KC $1/(p-1)$	KC $4/p(p-1)$
2	14.6	13.5	14.2	13.5
3	18.6	17.9	14.3	17.4
4	30.7	17.4	47.6	16.4
5	27.8	26.0	43.4	27.5
6	31.2	24.8	39.5	30.7
7	30.5	29.8	42.5	34.7
8	35.0	32.7	41.2	33.6
9	37.8	35.5	51.6	54.4

Table 2: Comparison of the two net models $\frac{1}{p-1}$ and $\frac{4}{p \cdot (p-1)}$ (best clusterings over embeddings in $\Re^d$, $1 \leq d \leq 5$) for AGG and KC. Results are for the Primary1 benchmark netlist and are evaluated using Scaled Cost [4].

Although Table 1 illustrates the superiority of AGG over KC, the best known implementation of AGG requires $O(n^2)$ time, while KC has an $O(n \log k)$ implementation. For the SKP problem, the speedup gained

Test Case	Number of Clusters - k (Best dimension)							
	9	8	7	6	5	4	3	2
19ks	15.0(10)	15.8(6)	15.6(1)	15.1(1)	14.4(1)	13.1(1)	12.5(1)	17.6(1)
bm1	27.6(6)	30.6(10)	28.6(10)	19.8(10)	17.9(6)	11.1(4)	7.0(2)	5.8(1)
Prim1	34.6(7)	33.6(5)	34.4(6)	30.7(4)	27.5(3)	16.4(3)	17.4(3)	13.5(1)
Prim2	11.7(6)	12.0(6)	11.8(9)	11.5(5)	10.4(4)	9.0(2)	7.5(1)	5.9(1)
Test02	21.5(8)	21.2(7)	21.1(6)	21.2(5)	23.1(3)	23.6(3)	19.1(2)	30.1(1)
Test03	21.0(9)	22.4(9)	23.2(9)	22.4(3)	22.2(2)	19.3(2)	21.4(2)	16.7(3)
Test04	22.1(6)	23.8(9)	24.4(6)	24.3(6)	27.2(5)	27.4(3)	36.0(2)	66.1(1)
Test05	11.0(6)	10.6(7)	10.7(5)	11.1(4)	10.3(10)	8.8(10)	10.2(6)	10.6(1)
Test06	31.0(10)	32.4(10)	33.6(4)	26.4(4)	28.8(4)	25.9(2)	19.3(2)	28.6(1)

Table 3: Scaled Cost measures of best k-way partitions obtained by KC using d-dimensional embeddings, $1 \leq d \leq 10$. Value in parentheses tells which dimension had the best Scaled Cost, illustrating the diagonal effect.

by KC is very significant. To facilitate comparison with future work by other researchers, Table 3 gives the best KC results over the geometric embeddings for $d = 1, \ldots, 10$, again measured by the Scaled Cost metric. We note that for a given value of k, the most favorable dimension d seems to grow roughly with k thereby creating a *diagonal effect* in the table.[5]

Finally, Table 4 shows comparisons between KC and the Shortest-Path Clustering (SPC) results given by Yeh et al., as measured by the Cluster Ratio objective [22]. In the table, we also include the "Recursive Ratio Cut" (RR) results that were reported in [22] (RSBipart is similar to RR; the former uses the spectral method of [10] while the latter uses the F-M variant of [20]). Here, as in [22], we assume uniform module areas. For each benchmark, and $2 \leq k \leq 9$, we report the best k-way partition obtained for $1 \leq d \leq 10$. Although our results are better than those of the recursive ratio cut approach, they are noticeably inferior to the SPC results. However, SPC is expensive in comparison with our method.

Test Case	RR		SPC		KC	
	cut(k)	R_C	cut(k)	R_C	cut(k)	R_C
19ks	11(2)	5.43	127(3)	4.72	32(5)	4.86
Prim1	11(2)	16.29	14(3)	12.81	14(2)	13.53
Prim2	83(2)	4.93	77(2)	4.61	88(2)	5.88
Test02	152(6)	16.62	9(3)	8.32	10(3)	8.72
Test03	48(2)	11.62	69(2)	10.71	52(3)	11.85
Test04	51(2)	11.54	6(2)	5.69	6(3)	5.85
Test05	48(2)	7.24	8(2)	3.11	15(4)	4.78
Test06	91(7)	7.23	81(9)	6.22	63(6)	9.34

Table 4: Results for KC, compared against those reported for Shortest-Path Clustering [23] according to the Cluster Ratio objective R_C (reported as a multiple of 10^{-5}).

[5] This seems to confirm the ideas of Chan et al., who also use k-dimensional embeddings to obtain k-way partitions; alternatively, this may also be an artifact of the objective function proposed in [4]. The "diagonal effect" implies that for any given value of k, only a few embeddings corresponding to $d \approx k$ need to be examined, and for k small, only a few eigenvectors need to be computed.

4.2 Results for the LKP Problem

Recall from the discussion of Section 1.2 that LKP solutions are typically evaluated by the improvement afforded to Fiduccia-Mattheyses bipartitioning, via the two-phase compaction approach. Here, we use the $O(n^2)$ AGG algorithm, since it is bottom-up and can be made fairly efficient for large values of $k = \Theta(n)$. Our motivation for this choice also stems from separate studies showing that AGG yields very high-quality solutions for Formulation 2 (min-diameter), especially when k is large.

In Table 5, we compare two-phase FM results using AGG clustering of the geometrically embedded netlist against the results for (i) standard F-M (on flat netlists), (ii) random matching-based clusterings, and (iii) RW-ST clusterings, all of which are quoted from [11]. For each benchmark, we use the same number of clusters k as was used in [11]; we report the best two-phase F-M result using agglomerative clustering in each of ten embedding dimensions ($1 \leq d \leq 10$) and 20 runs for each clustering (results of 200 F-M runs in parentheses). Our clusterings give an average of 26.9% improvement over the "flat" F-M partitioning results, in contrast to the 17.4% improvement given by RW-ST clusterings [11] using the same k values. Further, our best F-M value always occurred for $d \leq 3$. That low-dimensional embeddings are superior for LKP is interesting; recall that for SKP, the best embedding dimension grew with k.

Test Case	k	Standard FM Cuts	MBC Cuts	RW-ST Cuts	AGG Cuts
19ks	737	151 (140)	156	146	124
bm1	216	65 (61)	54	58	48
PrimGA1	191	66 (66)	48	47	49
PrimSC1	191	59 (59)	61	58	49
PrimGA2	702	242 (234)	187	165	146
PrimSC2	702	235 (235)	175	159	144
Test02	445	42 (42)	42	42	42
Test03	327	84 (84)	59	71	50
Test04	317	12 (12)	20	14	12
Test05	423	24 (24)	4	5	9
Test06	477	87 (65)	83	82	63

Table 5: Utility of AGG results within two-phase Fiduccia-Mattheyses partitioning.

5 Conclusion and Future Directions

We have introduced fast and effective k-way partitioning solutions for the two regimes $k \ll n$ and $k = \Theta(n)$. Our contributions include a new "partitioning-specific" net model used in constructing the Laplacian of the netlist, the use of fast Lanczos implementations to compute the d-dimensional geometric embedding efficiently, and a careful choice of geometric clustering objectives and algorithms. We believe that the spectral geometric embedding preserves important graph properties with respect to partitioning and clustering; moreover, the efficiency of geometric clustering opens the door for new heuristics.

We are pursuing several directions for future research. Since eigenvectors of the smaller eigenvalues correspond to better placements, *scaling* heuristics may be useful for perturbing the eigenvector-based embedding, e.g., we may compute the ratio of the i^{th}-smallest eigenvalue λ_i to the smallest eigenvalue λ_2, then reduce the components of the i^{th} eigenvector by this ratio. We believe that our studies of different net models and clustering algorithms represent promising directions. For example, the recent thesis of Zien [23] reports work of Chan, Schlag, and Zien which achieves excellent Scaled Cost results by combining a net model due to Frankle, a "partial KC" approach, and use of netlist information. This latest method of Chan et al. uses a variant of KC to make "easy" assignments to clusters; netlist connectivity information is then used to resolve more difficult points. By contrast, our present method relies only on the geometric embedding.

Finally, we hope to improve the running time of AGG since this method gives results uniformly superior to those of KC. Currently, the $O(n^2)$ method of [1] for general graphs is also the most efficient method known for geometric instances.

6 Acknowledgements

We are very grateful to the authors of [4] for providing early access to their works.

References

[1] J. P. Benzécri, "Construction d'une Classification Ascendante Hiérarchique par la Rechereche en Chaine des Voisins Réciproques", in *Les Cahiers de l'Analyse des Données* (VII)2, 1982, pp. 209-218.

[2] T. N. Bui, "Improving the Performance of the Kernighan-Lin and Simulated Annealing Graph Bisection Algorithms", in *Proc. ACM/IEEE Design Automation Conf.*, 1989, pp. 775-778.

[3] P. Brucker, "On the Complexity of Clustering Problems", in *Optimization and Operations Research*, 1977, pp. 45-54.

[4] P. K. Chan, M. D. F. Schlag and J. Zien, "Spectral K-Way Ratio Cut Partitioning and Clustering", to appear in *Proc. Symp. on Integrated Systems*, Seattle, March 1993.

[5] T. Feder and D. H. Greene, "Optimal Algorithms for Approximate Clustering", in *Proc. 20th Annual ACM Symp. Theory Computing*, 1988, pp. 434-444.

[6] C.M Fiduccia and R.M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions", in *Proc. ACM/IEEE Design Automation Conf.*, 1982, pp. 175-181.

[7] T. F. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance", in *Theoretical Computer Science*, 38, 1985, pp. 293-306.

[8] A. Guénoche, P. Hansen and B. Jaumard, "Efficient Algorithms for Divisive Hierarchical Clustering with the Diameter Criterion", in *Journal of Classification*, 8, 1991, pp. 5-30. 1990.

[9] S. W. Hadley, B. L. Mark and A. Vanelli, "An Efficient Eigenvector Approach for Finding Netlist Partitions", in *IEEE Trans. on CAD*, 11(7), July 1992, pp. 885-892.

[10] L. Hagen and A. B. Kahng, "New Spectral Methods for Ratio Cut Partitioning and Clustering", in *IEEE Trans. on CAD* 11(9), Sept. 1992, pp. 1074-1085.

[11] L. Hagen and A. B. Kahng, "A New Approach to Effective Circuit Clustering", in *Proc. IEEE Intl. Conf. on Computer-Aided Design*, Santa Clara, Nov. 1992, pp. 422-427.

[12] K.M. Hall, "An r-dimensional Quadratic Placement Algorithm", in *Manag. Sci*, 17, 1970, pp.219-229.

[13] J. Hershberger, "Minimizing the Sum of Diameters Efficiently" in *Proc. 3rd Canadian Conference on Computational Geometry*, 1991, pp. 62-65.

[14] S. C. Johnson, "Hierarchical Clustering Schemes", in *Psychometrika* 32(3), 1967, pp.241-254.

[15] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Wiley-Teubner, 1990.

[16] N. Megiddo and K. J. Supowit, "On the Complexity of Some Common Geometric Location Problems", in *Siam Journal of Computing*, 13(1), 1984, pp. 182-196.

[17] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer Verlag, New York, NY, 1985.

[18] L.A. Sanchis, "Multiple-way Network Partitioning", in *IEEE Trans. on Computers*, 38, 1989, pp. 62-81.

[19] D. G. Schweikert and B. W. Kernighan, "A Proper Model for the Partitioning of Electrical Circuits", in *Proc. Design Automation Conf.*, 1972, pp. 57-62.

[20] Y. C. Wei and C. K. Cheng, "Ratio Cut Partitioning for Hierarchical Designs", in *IEEE Trans. on CAD* 10(7), July 1991, pp. 911-921.

[21] C. W. Yeh, C. K. Cheng and T. T. Lin, "A General Purpose Multiple Way Partitioning Algorithm", in *Proc. ACM/IEEE Design Automation Conf.*, June 1991, pp. 421-426.

[22] C. W. Yeh, C. K. Cheng and T. T. Lin, "A Probabilistic Multicommodity-Flow Solution to Circuit Clustering Problems", in *Proc. IEEE Intl. Conf. on Computer-Aided Design*, Santa Clara, November 1992, pp. 428-431.

[23] J. Zien, "Spectral K-Way Ratio Cut Graph Partitioning", M.S. Thesis, Computer Engineering Dept., UC Santa Cruz, March 1993.