CBLO: A Clustering Based Linear Ordering for Netlist Partitioning

Kwang-Su Seong and Chong-Min Kyung
Department of Electrical Engineering
KAIST
Taejon, 305-701, Korea
Tel: 82-42-866-0700
Fax: 82-42-866-0702
e-mail: kyung@dalnara.kaist.ac.kr

Abstract— In this paper, we propose a clustering based linear ordering algorithm which consists of global ordering and local ordering. In the global ordering, the algorithm forms clusters from n given vertices and orders the clusters. In the local ordering, the elements in each cluster are linearly ordered. The linear order, thus produced, is used to obtain optimal k-way partitioning based on scaled cost objective function. Experiment with 11 benchmark circuits for k-way $(2 \leq k \leq 10)$ partitioning shows that the proposed algorithm yields an average of 10.6% improvement over MELO for the k-way scaled cost partitioning.

I. Introduction

A circuit netlist which is well represented as a hypergraph can be transformed to graph using the clique net model [3][7][9]. Because there are well-established theories and algorithms, many previous partitioning algorithms used the approximated graph model instead of a hypergraph[2][6]. The transformed graph $G(V, E)$ consists of vertex set, $V = \{v_1, v_2, \ldots, v_n\}$, and $n \times n$ symmetric adjacency matrix, $A = (a_{ij})$, where $a_{ij} > 0$ is the weight of $(v_i, v_j) \in E$. If no edge exists between vertex v_i and v_j, a_{ij} becomes 0. Here, we can define $n \times n$ degree matrix $D = (d_{ij})$, where d_{ii} is the degree of vertex v_i, *i.e.*, $d_{ii} = \sum_{j=1}^{n} a_{ij}$, and $d_{ij} = 0$ if $i \neq j$. The Laplacian of the graph G is defined as $Q = D - A$.

General definition of k-way partitioning problem is as follows: Given n modules represented as each element in the vertex set, $V = \{v_1, v_2, \ldots, v_n\}$, a k-way partitioning denoted as P^k is to divide n modules into a set of k disjoint clusters represented as $\{C_1, C_2, \ldots, C_k\}$.

Among many cost functions for measuring the quality of the result of partitioning [4][5][8], we chose to use the scaled cost function[4] to avoid the artificial size constraints. For a given hypergraph partitioning ,P^k, the scaled cost function is defined as follows;

$$F_{scaled}(P^k) = \frac{1}{n(k-1)} \sum_{l=1}^{k} \frac{E_l}{|C_l|} \quad (1)$$

where E_l denotes the number of signal nets crossing the boundary of cluster C_l, and $|C_l|$ denotes the size of cluster, *i.e.*, number of modules in C_l. The generalized ratio-cut cost function which is similar to the scaled cost function is defined as follows;

$$F_{ratio}(P^k) = \sum_{l=1}^{k} \frac{E_l}{|C_l|} \quad (2)$$

For a given graph partitioning P_g^k, the scaled cost function and the generalized ratio-cut cost function are the same as Eq. (1) and Eq. (2), respectively, except the meaning of E_l which denotes the sum of the weight of each edge crossing boundary of C_l, *i.e.*,

$$E_l = \sum_{v_i \in C_l, v_j \notin C_l} a_{ij}.$$

Spectral method uses the eigenvectors and eigenvalues of the Laplacian of a graph G. The vertices in the graph can be mapped into points in d-dimensional space using spectral method. Among many advancements made in this area[2][4], one-dimensional ordering scheme has received increasing attention[1]. Some partitioning algorithms have shown optimal results for a given order through the reduction of search space [3][7]. Thus, netlist partitioning problem can be transformed to linear ordering problem for the given vertices of a netlist.

EIG1[7] uses the second eigenvector of the graph's Laplacian matrix to place each vertex to one-dimensional space, which corresponds to a linear order. As a 2-way partition is obtained by cutting the linear order into two at one of $(n-1)$ cut points, EIG1 obtains the best 2-way ratio-cut partitioning among the $(n-1)$ partitionings. In SFC[3], d-dimensional eigenvectors are used. For given n points in d-dimensional space, SFC obtains a linear ordering using SFC(Space Filling Curve) which is a heuristic

ASP-DAC '97
0-89791-851-7$5.00

of Traveling Salesman Problem. MELO[2] considers n given points in d-dimensional space as vectors. Alpert et al. showed relation between graph partitioning and vector partitioning, which were utilized in MELO to construct a linear order by iteratively adding a vertex. The linear order produced by SFC and MELO are applied to DP-RP[3] for obtaining the optimal k-way partitioning with respect to the scaled cost function.

In this paper, we propose a new linear ordering algorithm which is then applied to DP-RP. It consists of three steps, *i.e.*, cluster formation, cluster ordering, and finally, ordering of elements within each clus ter. The cluster formation and ordering are also called global ordering, where the ratio-cut cost function is considered. On the other hand, the ordering of elements within each cluster is called local ordering.

The paper is organized as follows. Section II contains some preliminaries and the main algorithm proposed in this paper is described in section III. Experimental results are shown in section IV.

II. Preliminaries

For a given graph $G(V, E)$ and the corresponding $n \times n$ Laplacian matrix Q, the eigenvectors are denoted by $\vec{\mu_1}, \vec{\mu_2}, \ldots, \vec{\mu_n}$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ $(\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n)$. We assume that each eigenvector is normalized such that $\vec{\mu_i}^T \vec{\mu_i} = 1$ for all i. The $n \times n$ eigenvector matrix, U, has columns $\vec{\mu_1}, \vec{\mu_2}, \ldots, \vec{\mu_n}$, where each $\vec{\mu_i}$ is $n \times 1$ column vector. For some constant $H \geq \lambda_d$, we can define $n \times d$ matrix V_d consisting of scaled column eigenvectors follows;

$$V_d = [\vec{\mu_1}\sqrt{H-\lambda_1}, \vec{\mu_2}\sqrt{H-\lambda_2}, \ldots, \vec{\mu_d}\sqrt{H-\lambda_d}] \quad (3)$$

where $d \leq n$. Each vertex, v_i, in the graph $G(V, E)$ is mapped to a vector $\vec{y_i^d}$ in d-dimensional space, which denotes the i-th row of V_d [2]. The smallest eigenvalue $\lambda_1 = 0$ has corresponding eigenvector $\vec{\mu_1} = [\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}]^T$. As it does not contribute to partitioning, we can disregard the first column of scaled eigenvector V_d.[2]

Consider the problem of partitioning n vectors in d-dimensional space with the vector set represented as $Y = \{\vec{y_1^d}, \vec{y_2^d}, \ldots, \vec{y_n^d}\}$, where each graph vertex v_i corresponds to the vector $\vec{y_i^d}$. The definition of k-way vector partitioning P_v^k is to divide n vectors in Y into a set of k disjoint vectors $\{S_1, S_2, \ldots, S_k\}$. For a given vector partitioning P_v^k, we define a cost function as follows;

$$F_{ratio}(P_v^k) = \sum_{l=1}^{k} \frac{\|\vec{Y_l^n}\|^2}{|S_l|} \quad (4)$$

, where $\|\vec{Y_l^d}\|^2 = \|\sum_{\vec{y} \in S_l} \vec{y}\|^2$ and $|S_l|$ is the size of cluster S_l, *i.e.*, the number of vectors in S_l. We say that a graph partitioning solution $P_g^k = \{C_1, C_2, \ldots, C_k\}$ *corresponds* to a vector partitioning solution $P_v^k = \{S_1, S_2, \ldots, S_k\}$ if and only if $v_i \in C_l$ for every $\vec{y_i^d} \in S_l$. Alpert et al.[2] has proven the following relations to hold between vector partitioning and graph partitioning.
If $d = n$ and P_v^k corresponds to P_g^k, then

$$\sum_{i=1}^{k} \|\vec{Y_l^n}\|^2 = nH - \sum_{i=1}^{k} E_i \quad (5)$$

and

$$\sum_{i=1}^{k} \frac{\|\vec{Y_i^n}\|^2}{|S_i|} = kH - \sum_{i=1}^{k} \frac{E_i}{|C_i|} \quad (6)$$

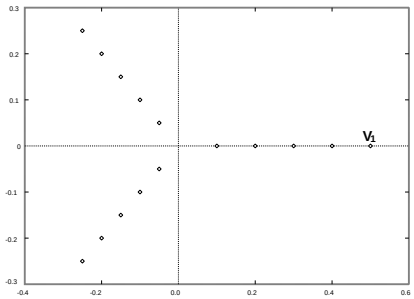
From the equations (4) and (6), we can say that if $F_{ratio}(P_v^k)$ is maximum, the corresponding graph partitioning, P_g^k, has minimum cost in terms of scaled cost and generalized ratio-cut objective function, *i.e.*, $F_{scaled}(P_g^k)$ and $F_{ratio}(P_g^k)$ are minimized. Thus, the minimization problem is converted to maximization problem.

III. CBLO: A clustering based linear ordering

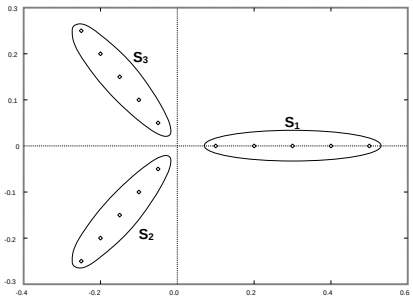
A. Motivation

As the proposed algorithm is based on the similar background as MELO[2], we investigate MELO before explaining the motivation of the proposed algorithm. If a cluster S maximizes $\|\sum_{\vec{y_i^d} \in S} \vec{y_i^d}\|^2$, the corresponding graph partitioning of S and $(Y - S)$ should be close to optimal in terms of 2-way min-cut partitioning. From the motivation, MELO[2] starts with a set of vectors S, which is initialized as an empty set, and iteratively adds to S the vector $\vec{y_i} \in (Y - S)$ that maximizes $\|\vec{y_i} + \sum_{\vec{y_j} \in S} \vec{y_j}\|^2$. The ordering scheme of MELO has a chance to produce good result in case of 2-way partitioning. But MELO, in general, is not proper for multi-way partitioning.

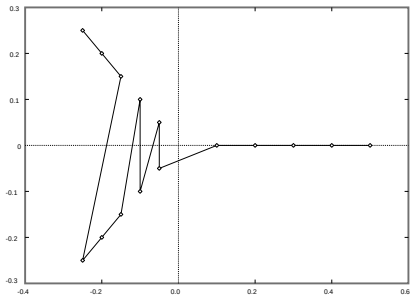
Let's take a look at the example shown in Fig. A. Here we assume that a vector set is given as Fig. A(a). Because the largest vector is v_1, MELO starts with v_1 and produces a linear order which is shown in Fig. A(c). It is difficult to find the optimal 3-way vector partitioning in terms of ratio objective, shown in Eq. (4), by cutting any two points in the produced linear order. The optimal 3-way vector partitioning is shown in Fig. A(b). In this paper, a clustering method is employed to improve the quality of multi-way partitioning. If a clustering method appropriately groups the vectors in d-dimensional vector space, we can improve the quality of linear ordering in multi-way partitioning. Fig. A(e) is the clusters and linear order produced by the proposed linear ordering



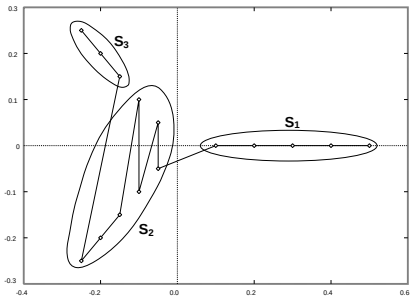
(a) An example: 15 vectors are plotted in 2-dimensional space.



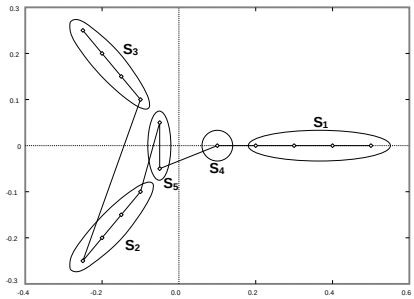
(b) Optimal 3-way vector partitioning in terms of ratio objective. $F_{ratio}(P_v^3) = 0.9$.



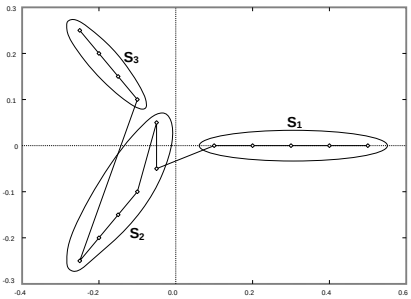
(c) Linear order produced by MELO



(d) 3-way vector partitioning using linear order produced by MELO. $F_{ratio}(P_v^3) = 0.857$



(e) 5 clusters and linear order produced by the proposed algorithm



(f) 3-way vector partitioning using the linear order produced by the proposed algorithm. $F_{ratio}(P_v^3) = 0.883$

Fig. 1. An example to show the effectiveness of the clustering based liner ordering algorithm

scheme, and Fig. A(f) is the result of vector partitioning which is closer to the optimal vector partitioning than the result of MELO shown in Fig. A(d).

B. Overview of the Proposed Algorithm

In the proposed linear ordering algorithm, n vectors, each representing n vertices respectively, are linearly ordered. The linear order, thus obtained, are later used as a basis for partitioning. The linear ordering process consists of two steps : global ordering and local ordering. The global ordering step again consists of two steps: cluster formation and inter-cluster ordering. In the cluster formation, vectors are grouped into an appropriate number of clusters such that vectors within each cluster are relatively strongly related with each other. The inter-cluster ordering is then performed to obtain the linear order of clusters. In the local ordering step, vectors within each cluster are linearly ordered.

C. Cluster formation

Conceptually, we form a cluster S_i by extracting vectors in the vector set Y, *i.e.*, a cluster S_i is formed through greedily merging the most attractive vectors. Now we will explain the detailed algorithm how to form a cluster S_i. The pseudo code of the cluster formation is shown in Fig. 2. In step (2.1), we choose a seed vector $\vec{y_s} \in Y$, which maximizes $\|\vec{y_s}\|^2$, and move it to the S_i. In step (2.2), we select a vector $\vec{y} \in Y$, which maximizes $\|\vec{Y_{S_i}} + \vec{y}\|^2$, where $\vec{Y_{S_i}} = \sum_{y \in S_i} \vec{y}$. If $F_{ratio}(S_i \cup \{\vec{y}\}) > F_{ratio}(S_i)$, *i.e.*, the ratio objective shown in Eq. (4), of S_i is increased by adding the vector to the set S_i, then we add the vector $\vec{y}$ to S_i, and go to the step (2.2). Otherwise, S_i is fixed as a cluster. The proposed clustering algorithm automatically decides the number of clusters and tries to maximize the ratio objective of the cluster S_i, *i.e.*, minimize the ratio-cut objective of the corresponding graph cluster C_i.

	The Cluster Formation
	Input : vector set $Y = \{\vec{y_1^d}, \vec{y_2^d}, \ldots, \vec{y_n^d}\}$
	Output: cluster set S
(1)	Set $i = 1$; S_1 = empty; Flag = 1;
(2)	**while** (Y is not empty)
(2.1)	choose a seed vector $\vec{y_s} \in Y$, such that $\|\vec{y_s}\|^2$ is maximal; move the vector to S_i;
(2.2)	**if** (Y is empty) **then** Flag = 0; **else** find a vector $\vec{y} \in Y$, which maximizes $\|\vec{Y_{S_i}} + \vec{y}\|^2$;
(2.3)	**if** (Flag = 1 and $F_{ratio}(S_i \cup \{\vec{y}\}) > F_{ratio}(S_i)$) **then** add the vector to S_i; go to step (2.2); **else** put the cluster S_i to the cluster set S; increment i; S_i = empty;

Fig. 2. The pseudo code of the cluster formation

D. Inter-cluster ordering

Let's assume that the number of clusters formed in the clustering step is q and the produced inter-cluster order is represented as $\{S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_q}\}$. The proposed inter-cluster ordering algorithm incrementally orders the given clusters. Let's define a segment, Δ, as a contiguous piece of the cluster order, *e.g.*, $\{S_{\pi_i}, S_{\pi_{(i+1)}}, \ldots, S_{\pi_j}\}$ can be a segment, where $1 \le i \le q$ and $i \le j \le q$. Initially each segment contains exactly one cluster and the inter-cluster order is obtained by merging two segments into one until only one segment remains. The remained segment has q ordered clusters, which can be represented as $\{S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_q}\}$. The pseudo-code of the cluster-wise ordering is shown in Fig. 4.

The criterion to select two segments is as follows. By merging two segments, we can get $q-1$ segments which is the optimum with respect to $F_{ratio}(P_v^{q-1})$ for a given q segments. If i-th and j-th segments are merged, then the corresponding $F_{ratio}(P_v^{q-1})$ is as follows;

$$F_{ratio}(P_v^{q-1}) = \sum_{l=1, l\neq i, l\neq j}^{q} \frac{\|\vec{Y_{\Delta_l}}\|^2}{|\Delta_l|} + \frac{\|\vec{Y_{\Delta_i}} + \vec{Y_{\Delta_j}}\|^2}{|\Delta_i \cup \Delta_j|} \quad (7)$$

$$= F_{ratio}(P_v^q) + \frac{\|\vec{Y_{\Delta_i}} + \vec{Y_{\Delta_j}}\|^2}{|\Delta_i \cup \Delta_j|} - \frac{\|\vec{Y_{\Delta_i}}\|^2}{|\Delta_i|} - \frac{\|\vec{Y_{\Delta_j}}\|^2}{|\Delta_j|} \quad (8)$$

Because q segments are already given, *i.e.*, $F_{ratio}(P_v^q)$ is fixed, we can obtain the optimum $(q-1)$ clusters in terms of ratio objective by choosing two segments which maximize the following objective.

$$F_{select}(\Delta_i, \Delta_j) = \frac{\|\vec{Y_{\Delta_i}} + \vec{Y_{\Delta_j}}\|^2}{|\Delta_i \cup \Delta_j|} - \frac{\|\vec{Y_{\Delta_i}}\|^2}{|\Delta_i|} - \frac{\|\vec{Y_{\Delta_j}}\|^2}{|\Delta_j|} \quad (9)$$

After two segments Δ_i and Δ_j are selected, they are merged into one segment according to the following procedure. Assume that segment Δ_i and Δ_j were obtained by merging Δ_{i1} and Δ_{i2}, and Δ_{j1} and Δ_{j2}, respectively. There are four different cases to merge two segments,

Fig. 3. Four different cases to merge two segments Δ_i and Δ_j.

which is shown in Fig. 3, where (a) and (b) show two different orientations for Δ_j. have been decided. In the merging step, we also try to maximize the ratio objective shown in Eq. (4) . If $F_{ratio}(\Delta_i \cup \Delta_{j1}) > F_{ratio}(\Delta_i \cup \Delta_{j2})$, we take the shape of Δ_j in case 1, and case 2 otherwise. The shape of Δ_i can then be decided by the similar procedure as the orientation of Δ_j has been fixed.

	The Inter-Cluster Ordering
	Input : q vector clusters $\{S_1, S_2, \ldots, S_q\}$.
	Output : ordered vector cluster $\{S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_q}\}$.
(1)	Initialize $M = \{\Delta_1, \Delta_2, \ldots, \Delta_q\}$ such that each segment contains exactly one cluster; Set m = q;
(2)	Find cluster Δ_i and Δ_j in M, which maximize cost function $F_{select}(\Delta_i, \Delta_j)$;
(3)	Merging Δ_i and Δ_j into new segment Δ_{new};
(4)	$M = (M \cup \{\Delta_{new}\}) - \{\Delta_i\} - \{\Delta_j\}$;
(5)	Decrement m; If $m > 1$ goto step 2;
(6)	The remained segment has ordered q clusters, which can be represented as $\{S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_q}\}$;

Fig. 4. The pseudo code of the inter-cluster ordering.

E. Local ordering

The motivation of the local ordering is the same as MELO[2]. Thus, if the size of each cluster is one, the proposed algorithm is exactly the same as MELO. The pseudo code of the local ordering is shown in Fig. 5. We will show how to order the vectors in the cluster S_{π_i}. Because the sequence of clusters for each of which local ordering is to be performed was already determined in

the inter-cluster ordering step, we assume that the vectors in each cluster S_l, $l < i$, was stored in the set O. The following procedure is repeated until all the vectors in the cluster S_i are ordered. A vector $\vec{y}$ which maximizes $\|\vec{Y_O}+\vec{y}\|^2$ is selected from S_i. The vector $\vec{y}$ is added to the set O and the corresponding vertex is labeled as v_{π_j} if $\vec{y}$ is the j-th vector added to the set O. Thus, we obtain the linear order of vectors in the cluster S_i. The above procedure is applied for all the clusters, then we can obtain the linear order of vertices $[v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}]$.

The Local Ordering

Input : vector set $\{S_{\pi_1}, S_{\pi_2}, \ldots, S_{\pi_q}\}$
Output : linear ordering $\{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}\}$

(1) O = empty;
(2) **for** $i = 1$ to q
  while (S_{π_i} is not empty)
    Find a vector $\vec{y} \in S_{\pi_i}$,
    which maximizes $\|\vec{Y_O} + \vec{y}\|^2$;
    Move the vector to O and label
    the corresponding vertex of the vector as v_{π_j}
    if $\vec{y}$ is the j-th vector added to the set O;

Fig. 5. The pseudo code of the local ordering

Time complexity of the proposed linear algorithm is $O(dn^2 + q^3)$, where q is the number of clusters. In the worst case, q is proportional to n. But in our experiment, the number of clusters is 16 in average case.

IV. Experimental Result

We implemented the proposed algorithm, called as CBLO, and compared with MELO and SFC for multi-way partitioning with the objective of minimizing the scaled cost function. The scaled cost function is widely used because it avoids the artificial size constraints and provides, as a single quantity, a measure of the quality of linear ordering[2].

Each netlist is transformed into the graph by clique model and a weight of w is assigned to each edge in the clique, where w is defined as

$$w = \frac{4}{p(p-1)} \frac{2^p - 2}{2^p}$$

After the hypergraph is transformed into the graph, eigenvectors were computed.

In this experiment, ACM/SIGDA benchmarks were used as test examples, which are shown in Table 1. The first column is test examples. The second, third and forth columns are the characteristics of benchmark circuits. The fifth column shows the number of clusters formed in clustering step. The sixth and seventh columns report Sun Ultra Sparc 1 runtime, whose unit is second, of proposed linear ordering algorithm and DP-RP, respectively.

Test Case	# of modules	# of nets	# of pins	# of clusters	runtime (CBLO)	runtime (DP-RP)
19ks	2844	3282	10547	15	6.6	110.6
prim1	833	902	2908	13	0.6	7.4
prim2	3014	3029	11219	13	7.2	126.0
test02	1663	1720	6134	17	2.4	32.8
test03	1607	1618	5807	19	2.1	31.2
test04	1515	1658	5975	20	1.8	27.5
test05	2595	2750	10076	18	5.3	95.2
test06	1752	1541	6638	10	2.9	38.3
balu	801	735	2697	17	0.5	6.6
struct	1952	1920	5471	12	3.1	47.0
biomed	6514	5742	21040	16	35.6	669.0
SUM	25090	24897	88512	170	68.1	1191.6

TABLE I
Benchmark list and statistics.

The runtime of proposed algorithm is to construct linear ordering after eigenvectors have been computed. We used 10 eigenvectors, $\vec{\mu_2}, \vec{\mu_3}, \ldots, \vec{\mu_{11}}$, to produce the data in Table 1 except characteristics of examples.

In the next experiment, we used d eigenvectors, where $2 \leq d \leq 11$, and then produced linear order for each value of d, *i.e.*, $d = 2$ uses eigenvector $\vec{\mu_2}$; $d = 11$ uses eigenvectors, $\vec{\mu_2}, \vec{\mu_3}, \ldots, \vec{\mu_{11}}$. When all the eigenvectors are not used it was shown that the selection of H affects the result of linear ordering. MELO proposed some scheme to select H. In MELO, it was shown that $H = \lambda_2 + \lambda_d$ yields slightly better solution. We also used the same H as the MELO in the experimental result.

The linear order produced through the proposed algorithm is then applied to the DP-RP[3] which produces optimal k-way partitioning with respect to the scaled cost function for a given linear ordering. In Table II, the proposed algorithm, CBLO, yields the best solution among those obtained by running DP-RP. The data in Table II except CBLO were obtained in MELO[2]. The proposed algorithm yields an average of 10.6% and 28.1% improvement over MELO and SFC in terms of the scaled cost function.

V. Conclusion and future work

In this paper, we proposed clustering based linear ordering for netlist partitioning. The proposed algorithm consists of global ordering and local ordering. The global ordering again consists of cluster formation and cluster based ordering. By clustering, the proposed algorithm has more global partitioning information than MELO, thus the proposed linear ordering scheme produces better results than MELO for multi-way partitioning.

Acknowledgments

We are grateful to the research group of UCLA for providing us with the MELO code, DP-RP code, benchmark examples and eigenvectors of the examples used in their work.

Test Case	Various algorithms	Number of Clusters,k									SUM
		10	9	8	7	6	5	4	3	2	
19ks	SFC	15.1	14.3	13.8	13.2	12.2	11.1	8.37	7.48	5.44	100.99
	MELO	9.85	9.10	8.31	7.87	6.84	6.14	5.32	4.99	4.79	63.21
	CBLO	8.92	7.87	6.90	6.30	6.09	5.89	5.50	5.08	4.79	57.34
prim1	SFC	38.9	36.7	35.2	31.7	28.8	26.0	21.8	14.6	13.4	247.1
	MELO	44.6	41.9	39.7	37.0	34.0	29.4	22.5	17.1	13.4	279.6
	CBLO	32.8	30.2	28.4	26.1	23.9	20.7	16.2	14.0	13.4	205.7
prim2	SFC	13.7	13.3	12.8	12.1	11.0	9.43	7.95	6.86	5.05	91.19
	MELO	13.7	12.7	12.0	11.2	10.1	9.18	7.95	6.76	4.71	88.30
	CBLO	11.5	10.9	10.0	9.68	9.05	8.40	7.67	6.42	4.64	78.26
test02	SFC	25.5	24.1	22.8	20.9	18.5	16.1	13.4	10.9	8.07	160.27
	MELO	21.1	19.9	18.5	17.0	15.4	13.9	12.4	10.7	8.07	136.97
	CBLO	20.0	18.8	17.6	16.1	14.5	13.0	11.7	10.5	8.07	130.27
test03	SFC	22.6	21.1	19.2	17.1	16.2	15.2	14.3	13.0	10.2	148.90
	MELO	19.0	17.6	16.7	15.3	14.6	13.7	12.5	11.6	9.25	130.25
	CBLO	17.4	16.5	15.8	14.9	14.0	12.7	11.3	10.3	9.19	122.09
test04	SFC	22.2	19.9	17.8	17.6	16.5	15.1	11.6	8.17	5.78	134.65
	MELO	13.2	12.3	11.5	10.8	9.97	9.32	8.21	6.83	5.78	87.91
	CBLO	13.6	12.5	11.2	10.2	9.53	8.65	7.39	6.45	5.70	85.22
test05	SFC	9.88	8.66	8.06	7.84	7.32	6.56	5.49	4.90	3.09	61.80
	MELO	7.42	7.03	6.53	6.11	5.79	5.50	4.85	4.35	3.09	50.67
	CBLO	6.58	6.26	6.02	5.71	5.15	4.72	4.37	3.96	3.06	45.83
test06	SFC	27.1	25.1	23.7	20.2	18.4	16.5	13.7	11.3	9.21	165.21
	MELO	21.3	20.2	18.5	16.7	14.7	13.5	11.3	9.54	8.80	134.54
	CBLO	19.4	18.3	16.8	15.1	13.5	12.4	11.6	9.92	8.15	125.17
balu	SFC	82.0	79.1	74.1	70.3	64.9	62.2	49.4	47.3	17.6	546.90
	MELO	54.0	50.1	46.5	43.2	40.0	36.7	32.3	24.4	17.6	344.80
	CBLO	49.7	44.4	42.3	40.4	37.7	34.3	28.1	25.6	17.7	320.20
struct	SFC	12.1	11.2	10.5	9.41	8.65	7.93	7.05	6.42	4.85	78.11
	MELO	12.9	12.0	10.9	9.82	8.46	7.56	6.53	5.54	4.25	77.96
	CBLO	11.8	11.0	10.0	9.29	8.60	7.95	7.43	5.93	4.70	76.70
biomed	SFC	1.84	1.69	1.59	1.47	1.51	1.48	1.25	1.15	0.85	12.83
	MELO	1.87	1.73	1.62	1.49	1.34	1.23	1.11	0.89	0.61	11.89
	CBLO	1.63	1.54	1.43	1.29	1.20	1.08	0.94	0.89	0.61	10.61
SUM	SFC	270.92	255.15	239.55	221.82	203.98	187.60	154.31	132.08	83.54	1748.95
	MELO	218.94	204.56	190.76	176.49	161.20	146.13	124.97	102.70	80.35	1406.10
	CBLO	193.33	178.27	166.45	155.07	143.22	129.79	112.20	99.05	80.01	1257.39

TABLE II
COMPARISONS OF THE PROPOSED LINEAR ORDERING ALGORITHM WITH MELO AND SFC FOR 11 TEST CIRCUITS WITH VARIOUS NUMBER OF CLUSTERS, k, IN TERMS OF THE SCALED COST FUNCTION WITH THE SCALING FACTOR OF 10^{-5}

REFERENCES

[1] C.J.Alpert and A.B.Kahng, "Recent directions in netlist partitioning: a survey," INTEGRATION, the VLSI journal, 1995, 19 ,pp.1-81.

[2] C.J.Alpert and S.-Z.Yao, "Spectral Partitioning:The More Eigenvectors, the Better," Proc. ACM/IEEE Design Automation Conf., 1995,pp.195-200.

[3] C.J.Alpert and A.B.Kahng, "Multi-Way Partitioning Via Spacefilling Curves and Dynamic Programming," Proc. ACM/IEEE Design Automation Conf., 1994,pp.652-657.

[4] P.K.Chan, M.D.F.Schlag and J.Zien, "Spectral K-Way Ratio-Cut Partitioning and Clustering," IEEE Trans. on CAD 13(9), 1994,pp.1088-1096.

[5] C.J.Alpert and A.B.Kahng, "Geometric Embeddings for Faster and Better Multi-Way Netlist Partitioning," Proc. ACM/IEEE Design Automation Conf., 1993,pp.743-748.

[6] B.M.Riess, K.Doll and F.M.Johannes, "Partitioning Very Large Circuits Using Analytical Placement Techniques," Proc. ACM/IEEE Design Automation Conf., 1994,pp.646-651.

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

[8] C.J.Alpert and A.B.Kahng, "A General Framework for Vertex Ordering, With Applications to Netlist Clustering," Proc. IEEE/ACM Int. Conf. on Computer-Aided Design,Nov.1994, pp.63-67.

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

[10] D.S.Scott,"LASO2 Documentation", technical report, CS Dept., University of Texas at Austin 1980.