Unimodular Hypergraph based clustering approaches for VLSI Circuit Partitioning

R.Manikandan[1] , P.Swaminathan[2] and R.Sujitha[3]
Senior Assistant Professor, Information and Communication Technology[1]
SASTRA University, Thanjavur- 613 401, Tamilnadu, India.
Dean,School of Computing[2]
SASTRA University, Thanjavur- 613 401, Tamilnadu, India.
PG Student, VLSI Design [3]
SASTRA University, Thanjavur- 613 401, Tamilnadu, India.
(Email: manikandan75@core.sastra.edu)

Abstract-The reliability of Integrated circuits mainly depends on the VLSI Design Automation techniques. The main problem faced in these techniques is the Physical Design Implementation. The size and performance parameters of an integrated circuit mainly rely on the three main domains of Physical Design which are Placement, Partitioning, and Routing. In this paper, the problem domain is expressed in Unimodular Hypergraph. Comparison between Two-Step cluster method, Divisive-Hierarchical Cluster technique and the K-medoid cluster technique is analyzed. The Unimodular Hypergraph is used to obtain the maximum flow at a particular node and also to minimize the time between to nodes.

Keywords: Unimodular Hypergraph, Two-Step Cluster, Hierarchical Cluster technique, K-medoid cluster technique

1.Introduction

Division of a circuit or system into a collection of smaller parts (components) is referred as Partitioning. It is a design task to break a large system into pieces to be implemented on separate interacting components. It serves as an algorithmic method to solve difficult and complex combinatorial optimization problems as in logic or layout synthesis. The enormous increase of system complexity in the past and the expected further advances of microelectronic system design and fabrication are main reasons for partitioning to become a central and sometimes critical design task.

The complexity of the circuit has become so high due to the increase in the size of VLSI designs that it is very difficult to simulate the whole system without decomposing it into sets of smaller subsystems. This divide and conquer strategy relies on partitioning technique. The main challenge of the VLSI Physical design is to place the million of transistors on a silicon wafer, connect them with wires and make it work in perfect synchronization. Hypergraph partitioning within VLSI Physical design is a reliable longstanding application in Clustering methods.

Clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. It can be achieved by various algorithms that group the nodes with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions etc. It is a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Clustering has wide applications that include data-mining. It is a statistical technique for data analysis used in machine learning, pattern recognition, information retrieval and bio-informatics.

The Hypergraph partitioning problem most commonly arises in the context of dividing a circuit specification into clusters of components, such that the cluster interconnect is minimized. Each subcircuit can then be assembled independently, speeding up the design and integration processes. A circuit specification includes *cells*, which are pre-designed integrated circuit modules that implement a specific function and have input and output terminals. A Hypergraph is used to represent the connectivity information from the circuit specification.

Hierarchical clustering is a method to build a hierarchy of clusters. It is mainly of two categories: Agglomerative and Divisive. The former is a “bottom up” method whereas later is the “top down” approach. In Agglomerative, each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. In Divisive, all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. The *k*-medoids clustering algorithm attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the *k*-means algorithm, *k*-medoids chooses centers for the representation of the cluster. Here, a comparative study between two-step hierarchical cluster, divisive hierarchical clustering and K-medoid cluster technique is done using Hypergraph. Section 2, 3, 4 details about the two-step, hierarchical, k-medoid cluster techniques. Section 5 briefs about unimodular hypergraph.

2.Two-Step Clustering

Two step clusters is an algorithm that is mainly designed for the analysis of larger datasets. Many features of Two-step clustering methods differentiate it from traditional clustering algorithms. These include the ability to handle both continuous and categorical variables, to automatically determine the optimum number of clusters and to handle very large datasets,. The Two-step clustering is based upon a sequential approach of creating clusters and then sub-clusters.

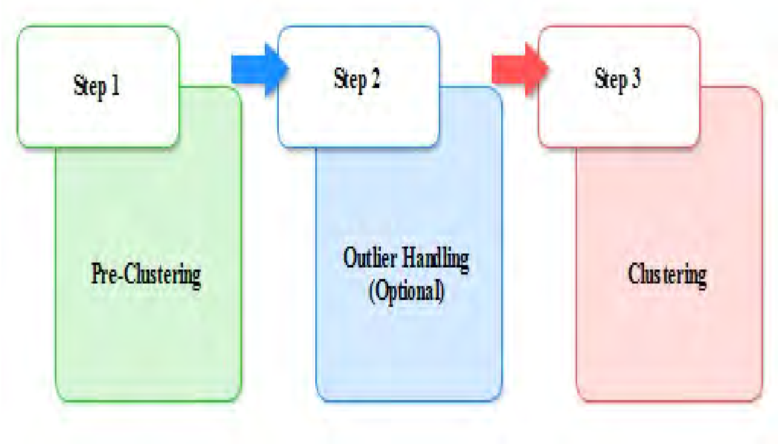


Figure 1: Two-Step Clustering.

The name 'Two-step clustering' is indicates that the method is based on a two-stage approach: Firstly, the algorithm undertakes a procedure that is very similar to the K-means algorithm. Secondly, the algorithm conducts a modified Hierarchical agglomerative clustering procedure by combining the objects sequentially to form homogenous clusters. This is done by building a so-called cluster feature tree whose "leaves" represent distinct objects in the dataset.

3 Hierarchical Clustering

The Partition Algorithms mainly specify the number of partitions and then iteratively relocate the nodes between the partition sets depending on the notion of the particular algorithm. On the other hand, the Hierarchical Clustering algorithm divides the existing groups or merges two or more groups, thus creating a hierarchical structure which reflects the order in which the division or merging has taken place.

It has mainly two subcategories.

- Agglomerative
- Divisive

The pros of choosing Hierarchical clustering are its versatile nature, ease of handling any forms similarity and distance and its embedded flexibility regarding the level of granularity.

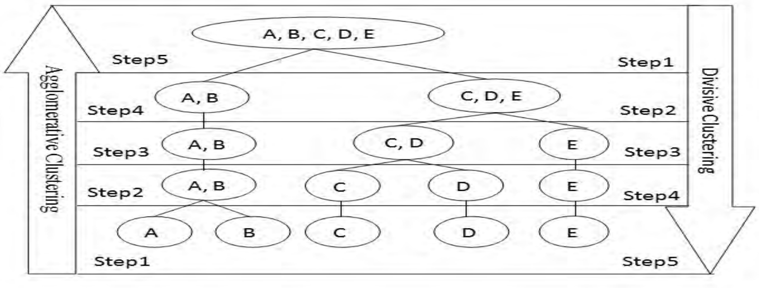


Figure 2: Two methods of Hierarchical Clustering

In Agglomerative, each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. In Divisive, all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

4. K-medoids Clustering

The K-means technique uses the centroid for representing the cluster and it is highly sensitive to a node that lies far from the observant (outliers). This issue is overthrown by using K-medoid Cluster technique. This method uses medoids, instead of centroid, for representing the cluster. A medoid is the most centrally located node in a cluster.

Here, k data objects are selected randomly as medoids to represent k cluster and remaining all data objects are placed in a cluster having medoid nearest (or most similar) to that data object. After processing all data objects, new medoid is determined which can represent cluster in a better way and the entire process is repeated. Again all data objects are bound to the clusters based on the new medoids. In each iteration, medoids change their location step by step. In other words, medoids move in each iteration. This process is continued until no any medoid move. As a result, k clusters are found representing a set of n data objects.

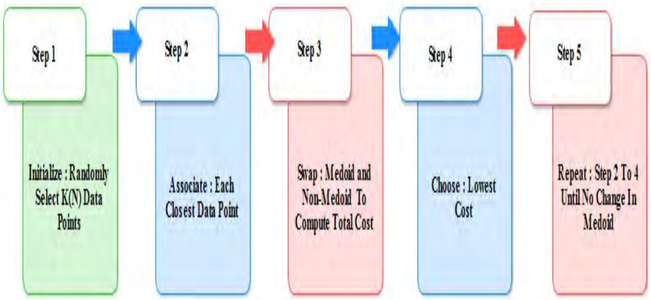


Figure 3:K-Medoids cluster.

5.Unimodular Hypergraph

A graph G is a unimodular graph if the incidence matrix between the vertices and the maximal cliques of G is totally unimodular. A hyper graph K is a unimodular hyper graph
 if the vertex-edge incidence matrix of K is totally unimodular. The unimodular Hypergraph not only determines the maximum flow at a particular node but also states the timing parameter between the nodes.

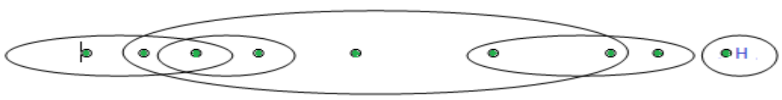


Figure 4: Unimodular Hypergraph.

$$H = \{[n], \{i, \ldots j\} / \ 1 \le \mathrm{i} \le \mathrm{j} \le \mathrm{n}\}$$

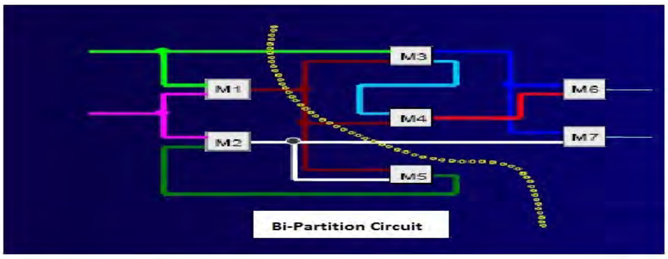
$$Disc(H) \le 1, |E| \, even$$

$$X(E) = 0, |E| \, odd \ \text{ and Perfectly balanced}$$

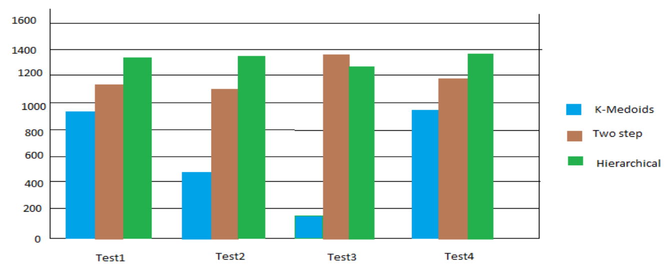
$$|X(E)| = 1 \text{ atmost perfect 1 cannot be avoided}$$

6.Results and Analysis

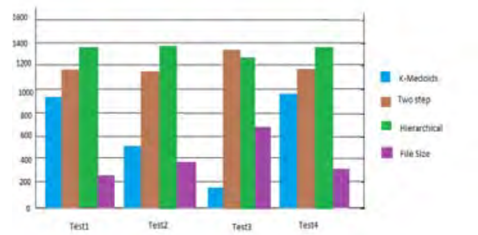
The Clustering Algorithms were compared based on the size of data, Quality of the clustering, Timing factor. For each factor, four tests were conducted: one for each algorithm.



Run Time Performance:



Data Analysis:



7.Conclusion

The evaluation conducted in this paper proves that K-medoids algorithm gives better result. It takes less time for partitioning of VLSI Circuit . Quality analysis and data analysis point of view K-medoids algorithm is well suited for VLSI Circuit partitioning . Minimum time bound between two nodes and maximum flow at a node were achieved through Unimodular hypergraph representation.

References

[1] R.Manikandan,P.Swaminathan, R.Sujitha"A Study on Vlsi Physical Design Specific Issues", Journal for Applied Sciences and Research 9(1):100-103,2013.

[2] Sumitra Devi, K.A., N.P. Banashree and Annamma Abraham, 2007. "Comparative study of Evolutionary model and Clustering methods in circuit partitioning pertaining to VLSI Design", World Academic Sciences, Engineering and Technology, 2: 42-45.

[3] Osama Abu Abbas,"Comparison between Data Clustering Algorithms", The International Arab Journal for Information Technology. Vol.5 No.3. July 2008.

[4] Shalini S Singh, N C Chauhan , " k-Means v/s k-Medoid :Comparative Study", National Conference on Recent Trends in Engineering & Technology.

[5] R.Manikandan, "Improving Efficiency of Textual Static Web Content Mining using clustering techniques", Journal for Theoretical and Applied Technology. Vol.33 No.2.