International Journal of Trend in Research and Development, Volume 4(1), ISSN: 2394-9333
www.ijtrd.com

A Survey on Hypergraph Partitioning Techniques

[1]Akanksha J. Kulkarni and [2]Swati Bhavsar,
[1]ME Student, [2]Assistant Professor,
[1,2]Computer Engineering Department, Matoshri College of Engineering & Research Centre, Eklhare, Nashik, India

Abstract**—**Basically Graphs can be used to store any kind of data in the form of vertices and edges. Most of the online structures used graphs to store & model this data. So Graph Partitioning is a strategy to divide a given graph says "Hypergraph" into its smaller sub parts. These smaller parts are nothing but sub-graphs of the original graph. This sub graphs represent the efficient and highly optimized structure of the original Hypergraph. This paper mainly focuses on various techniques used earlier to divide "Positive Hypergraph" into its dense sub-graphs.

Keywords**—***Positive Hypergraph, Graph Partitioning*

I. INTRODUCTION

Graph Partitioning is a problem of dividing original Hypergraph G= {V, E} into its equivalent sub graphs which will represent dense, efficient and optimized structure of the Hypergraph. Positive Hypergraph is a graph whose edges having the positive weights but self loops can be negative.As we are aware with the fact that, any kind of data or information can be modelled by a graph by transforming it into a graph structure such as G= {V, E}, where V represents set of vertices, which can be act as basic entities performing the roles, E represents set of edges act as the connections between the entities. Since we can store data in the form of graphs, then retrieving& storing, data structure used adjacency matrix/lists. But then the question arises is that what is the need to partition those graph structures?

Let us consider graph structure G for say "Facebook", where one node/vertex represents 1 person on Facebook & its association with the other vertex.

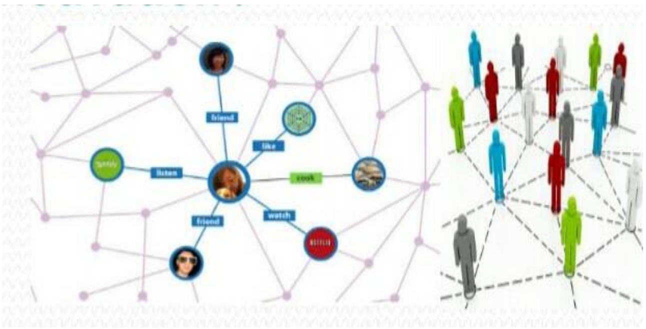

Fig 1: Connected Structure of Graph G= {V, E}

However this can extend in a way like, connection between say vertex A and B then B to any of its friend, that friends connection might be with any other vertex and so on. So it results in complex, crowded structures of vertex and edges and websites are burdened to manage & operate such a high volumes of data. But partitioning strategy will divide such a high volumes of graph into its smaller sub-graphs for efficient computing by reducing memory and space requirement for storing such a high volumes of data.

Various schemes have been proposed earlier for partitioning the original graph into sub-graphs. Various softwares are also used for the partitioning purpose.

Ker-nighan Lin Algorithm undergoes partitioning problem, that focuses on partition of the graph having nodes with weights on the edges into the sub graphs of given sizes which are equal such that weights on all the edges will be minimised. This Approach results in an optimal partitions and it is also very fast that can be employed to solve large problems.

K-way Maximum Sum of Densities method decomposes a Hypergraph into k parts, such that the sum of the densities of all k parts is to be maximized.

K-means method is one of the popular method used for the clustering, but it has some disadvantages is that it can't distinguished the clusters which non-liner separable in the input. To solve this problem spectral partitions [10] are used based on eigenvectors of the affinity matrix, with the motive to minimize the cut.

Net-Flow based partition methods are basically max flow min cut theorem. Ford and Fulkerson has proposed partitioning method called as 'Max Flow Min Cut' theorem.

Felzenszwalb and Huttenlocher has been proposed a method for image segmentation, in which image is partitioned into the regions by representing the images as the graph with the use of the predicate.

Softwares such as 1) METIS 2) JOSTLE 3) SCOTCH 4) CHACO are also used for partitioning Positive Hypergraph into Dense Subgraph.

Study of such previous techniques and softwares used is summarized in next few sections in which techniques are divided as Hypergraph Partitioning Methods and Dense Subgraph Detection Methods.

II. HYPERGRAPH PARTITIONING METHODS

A. Ker-nighan Lin Algorithm

This algorithm[1] mainly deals with the problem that, Graph G is given with edges having weights of it, and problem is to divide the graph G into sub graphs which should not exceed given maximum limit and thereby it will achieve minimal cost for edge cut. This Kernighan Lin Algorithm frames the partitioning problem mathematically such as there is graph G with n vertices and weights on the edges is say w, provided that w>0.

But the procedure for finding partition with minimum cost is difficult. This algorithm uses the given graph of 2n vertices and partitions it into 2 sub graphs of n size each. That means Kernighan Lin algorithm partitions the graph into 2 equal size sub graphs. Goal is to minimize the external cost which is the summation of cost of partition A and partition B having exactly n pints each. The method used was, initially they have started from any partition A, B of original graph G and made an attempt to reduce the external cost by making number of Updations to sub graphs A and B. When no more Updations are possible, then resulted partition can be considered to be minimal.

B. K-way Maximum Sum of Densities Method

K-way Maximum Sum of Densities method [2] decomposes a Hypergraph into k parts, such that the sum of the densities of all k parts is to be maximized. It uses g a peeling-off strategy, in which this method iteratively finds sub graphs of maximum densities of the original graph G and obtains particular order of the vertices. Then after applying dynamic programming on this order original graph G will be decomposed into k.

C. Max Flow Min Cut Method

Net-Flow based partition methods are basically max flow min cut theorem[1].Ford and Fulkerson has proposed partitioning method called as ’Max Flow Min Cut’ theorem, in that graph was treated as a network and weights on the edges were considered as max flow capacity between any 2 vertices. This approach states that max flow values should be equal to the minimum cut capacity of all the cuts. So the total cost of the partition can be considered as cut capacity.So this method gives the cut with the maximal flow which in turn results in a minimal cost. This algorithm will further resulted in the drawback that there is no any way to specify or to limit sizes of the sub graphs to be generated. That means if net flow methods are to be used for solving the partitioning problem, then appropriate processing is required to output sub graphs of the desired size. If the sub graphs obtained differ in sizes, then benefit of the algorithm will be lost.

D. K means Method

One of the widely used methods for clustering is the K-means method. One of the disadvantage of this method is it is not capable of disconnecting clusters that are non-linearly distinguishable. To resolve this, spectral partitions [3] are used based on eigenvectors of the matrix, by making an attempt to minimise the cut.

Another way to resolve this, kernel k means in which pints are first mapped and then it segments the points by using linear separations.

Basically kernel k -means and spectral clustering seems to be different. k -means objective function can be generalised so that it can make use of weights and kernels. However k means objective function can be rewrite as a trace maximization problem, in which eigenvectors are used to tackle its relaxation.

By choosing the weights in certain order, objective function of the kernel k means can be similar to the normalized cut. Thus the algorithms that based on Eigen vectors are used to minimise normalized cuts in spectral clustering and image segmentation.

However, software is required to calculate Eigen vectors for large matrix results in a greater overhead. This situation arises when large number of Eigen vectors need to be used. In such situations, Kernel k means can be used iteratively so that it will minimize the cut for a graph G directly.

E. Image Segmentation Method

Felzenszwalb and Huttenlocher has been proposed a method for image segmentation [4], in which image is partitioned into the regions by representing the images as the graph with the use of the predicate. This predicate will provide certain measurements of the separation or boundary between the regions by representing the image as a graph.

This approach based on Graph based approach. Such that Graph G = {V, E}, where V represents set of elements to be segment and E represents 2 consecutive neighboring vertex. Each edge ∈ E associated with certain non-negative weight which shows difference between vertices connected by that edge. For example, for any image vertices will be pixels of that image and difference might be in colour, intensity, motion etc.

In this approach quality or effectiveness of this partitioning can be measured by checking whether the elements in the parts of the segmentation are equivalent and elements in different part not to be similar. This means that edges connecting to the pair of vertices within the same partition should have low weights and edges connecting to the vertices in different partitions should have high weight.

This method makes use of predicate for checking the proof for the boundary between two partitions of segmentation. Building of this predicate is based on calculating the difference between elements which resides along the boundary between the partitions. This predicate then compares the partition to partition differences with within partition differences.

Time complexity for this approach depends on the implementation steps, as in fist step all the edges will be sort in non-decreasing order. And for positive integer weights it requires linear time O (m log m) time. To check whether any 2 vertices are in the same partition, uses set find for each vertex. And to merge two partitions uses set-union function. Minimum Internal Difference can be calculated in normal time if internal difference and size of the partition is known.

F. Other Methods

There are many other methods like multicut method for planar graph in [5] that results in optimal partitions, classic heuristic method proposed by Fiduccia-Mattheyses algorithm [6] that separates original Hypergraph G in two parts under certain area ratio with the objective is that the cut should be minimized.

Multilevel partition methods [7] will results for both the benefits such as quality of the partition and reducing the computational burden. Original graph G is first simplified then decomposed into its sub graphs and then updated or fine tune the resulted partitions for proper and efficient results.

With the point of view of Edge Separation, Bansal et al. proposed the correlation clustering method [8] that decomposes binary graph into its sub graphs automatically. Emanuel and Fiat used this concept for weighted graphs. Kim et al. further used this method for the Hypergraph for the betterment in the task of image segmentation [9].

III. DENSE SUBGRAPH DETECTION METHOD

Due to the much more importance of dense sub graphs there is an algorithm for finding large, dense sub graphs in massive graphs. This algorithm is based on application of fingerprinting by using shingles in a recursive manner. Basically this algorithm has been applied to large dense sub a graph that gives the connections between say users of www web.

IV. SOFTWARES

Various Softwares are available for partitioning Positive Hypergraph into dense sub graphs. Some of them are described below:

A. METIS

METIS [10] is software used for Serial Graph Partitioning and Fill-reducing Matrix Ordering. METIS consists of set of serial programs used for dividing the graphs. The algorithms which are implemented in METIS are based for multilevel partitioning techniques.

METIS resulted in a partition set for given graph G which better than any other algorithms used. Also when compares with Spectral Partitioning Schemes, METIS is better to provide results 10-50% better than spectral techniques.

Computation performed in METIS is very fast one to two orders of magnitude faster than any other technique. For example if a graph G has several vertices then also it can be divided into 256 pats in a very less amount of time. METIS basically aims at lesson down the storage and computation requirements.

B. hMETIS

hMETIS softwares mainly used for partitioning the Hypergraph and VLSI circuits. hMETIS is a set of programs for partitioning Hypergraph as to VLSI circuits. The algorithms used in hMETIS are based on the multilevel Hypergraph partitioning techniques.

As hMETIS is aimed at partitioning the VLSI circuits, it produces partitions that cuts 10-300% hyperedges when compares with other algorithms such as PARABOLI, PROP and CLIP-PROP.

C. SCOTCH

SCOTCH [11] is mainly aims at applying the graph based theory that uses divide and conquer strategy to solve various computing problems such as graph partitioning, static mapping etc.

The **SCOTCH** consists of set of programs and useful libraries which performs the static mapping and sparse matrix reordering algorithms in the SCOTCH. It offers both C and FORTRAN interfaces. SCOTCH provides variety algorithms for partitioning graphs and mesh structures.

As the SCOTCH provides static mapping, it has an ability to map source graph which is weighted to any target graph. This allows programs to be mapped onto the partitions of architectures which might be disconnected and made up of connection links and heterogeneous processors.

Time required for SCOTCH to perform partition or computations is linear to the number of edges in the graph, that means as the edges will increase or decrease time will also be increased or decreased. However time is logarithmic to number of vertices in the given graph G.

Large graphs can also process well in the SCOTCH with an attempt to utilize memory very efficiently without resulting in any out of memory errors. Since the SCOTCH has released under CeCILL-C free/libre software license, it can effectively used as testing platform for development of the new partitioning strategies and testing of those techniques.

SCOTCH provides variety of tools for generation checking and displaying large graphs. A SCOTCH program written in C and it is based on POSIX interface, which make them very portable.

D. JOSTLE

JOSTLE [12] is software used for partitioning the multilevel graphs which are parallel.

JOSTLE uses multilevel partitioning strategy, which usually match and joins neighbor vertices which will help to build new graphs. This process is repeated until the graph size lesson down to some threshold. The coarse graph is then divided and resulting partition will updated on all the graphs starting with coarse graph and ending with original graph G. At each and every step in which changes occurs, final division of coarse graph is used as an input to build starting partition for the next level.

Following figure shows the example of how multilevel partitioning technique is applied to a graph. Starting from left the original graph G is reduced to 4 vertices which then divided into 4 sets of partitioning by applying the refinement on those partition sets with an attempt to provide high quality partition sets for a given graph G.

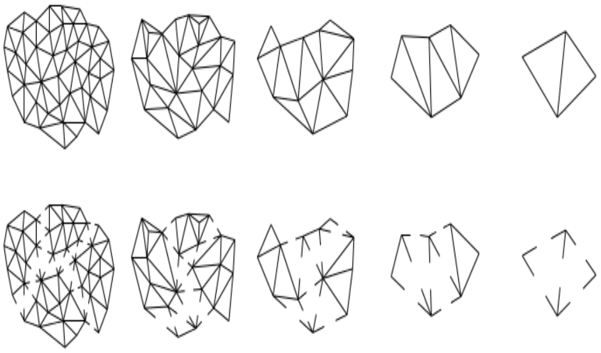

Fig 2: Multilevel Partitioning

Mesh or Graph Partitioning problems basically occur in one of the following manner:

It can be either static partitioning problem, which makes an attempt to distribute given graph or mesh across processors, or it can be static load balancing problem which occurs from a graph which is generated in parallel way, or it can be dynamic load partitioning problem, which occurs from refined graphs or the graphs in which workload changes linearly with the time.

In static and dynamic load balancing problem, input is distributed graph which is not load balanced and also not optimal. So this problem can be deals by shifting the graph on one of the processor and then it will responsible for implementing serial partitioning algorithm and distributes the graph again. But again it is not a good approach has disadvantages. As O (N) can cause overhead is that it fails to be scalable if the entity that is responsible for solving the problem is running at O (N/P), N is no. of vertices in graph and P is the no of processors available in the system.

Another problem is that graph cannot be fit into the memory of the machines and so leading to the delays in the memory paging operation.

Also when an optimal partition set already exists then it requires to reuse this set as an inception for partitioning it again.

So by considering all these issues JOSTLE provides the partitioning bench such that,

1) It has an ability to work in parallel.
2) It has ability to optimized existing partition.
3) It has an ability to discover partition sets with good features.
4) It also has an ability to balance the workload well.

CONCLUSION

So this paper focused on the study of all the previously used techniques for graph partitioning such as Kernighan-Lin Algorithm, Multilevel Partitioning, Max Flow Min Cut theorem in the categories of Hypergraph Partitioning Methods and Dense Subgraph Detection Methods. This paper also focuses on various softwares available for partitioning such as METIS, hMETIS, SCOTCH, JOSTLE along with their diverse effects.

Acknowledgment

Inspiration and guidance are invaluable in every aspect of life, especially in the field of education, which I have received from our respected H.O.D. Dr. Varsha H. Patil who has guided me in this work and gave earnest co-operation whenever required. I would like to express my sincere gratitude towards her.

I am pleased to announce that this work would not have been completed without the able guidance and complete support of Prof. S.A. Bhavsar who helped me at each and every step in every possible way. She always provided me with access to the latest technology and facilities and encouragement at every point and took active participation in the achievement of my objective. Heartfelt my foremost thanks go to seminar guide and help of my well-wishers and colleagues.

At last, I would like to take this opportunity to convey thanks to SPPU, all my staff members, who directly or indirectly encouraged and helped me to complete my work on time and contributed their valuable time in helping me to achieve success in this work.

References

[1] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," Bell Syst. Tech. J., vol. 49, pp. 291–307, 1970.

[2] D.-H. Huang and A. B. Kahng, "When clusters meet partitions: New density-based methods for circuit decomposition," in Proc. Eur. Conf. Des. Test, 1995, pp. 60–64.

[3] I. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in Proc. ACM Int. Conf. Knowl. Discov. Data Min., 2004, pp. 551–556.

[4] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based

Image segmentation," Int. J. Comput. Vis., vol. 59, no. 2, pp. 167–181, 2004

[5] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schn, "Globally optimal image partitioning by multicuts," in Proc. Energy Minim. Methods Comput. Vis. Pattern Recognit., 2011, pp. 31–44.

[6] C. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions," in Proc. 19th Conf. Des. Autom. 1982, pp. 175–181.

[7] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," SIAM J. Sci. Comput., vol. 20, no. 1, pp. 359–392, 1998.

[8] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," Mach. Learn., no. 1-3, vol. 56, pp. 89–113, 2004.

[9] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo, "Higher-order correlation clustering for image segmentation," in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 1530–1538.

[10] http://glaros.dtc.umn.edu/gkhome/views/metis/index.html

[11] http://www.labri.u-bordeaux.fr/perso/pelegrin/scotch/

[12] http://staffweb.cms.gre.ac.uk/~c.walshaw/jostle/