ENHANCED MULTILEVEL HYBRID ALGORITHM FOR GRAPH PARTITIONING

A Dissertation submitted

By

Annu Arora

to

Department of

In partial fulfilment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science and Engineering

Under the guidance of

Assistant Professor
Karanvir Kaur

(April 2015)

PAC APPROVAL PAGE

INDIA'S LARGEST UNIVERSITY*

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the student : Annu Arora
Batch : 2013
Session : 2014-2015

Registration No : 11304935
Roll No : RK2305B38
Parent Section : K2305

Details of Supervisor:

Name : Karanvir Kaur
UID : 14856

Designation : Assistant Professor
Qualification : M.E
Research Exp. : 4 years

Specialization Area: Database (pick from list of provided specialization areas by DAA)

Proposed Topics:-

1. Design and implementation of algorithms in data structures.
2. Comparative study of data sets in DBMS.
3. Clustering techniques in data mining.

Signature of supervisor 14856

PAC Remarks: Topic 1 is approved.

11058

APPROVAL OF PAC CHAIRMAN Signature: Date: 24/11/14

*Supervision should finally encircle one topic out of three proposed topics and put up for an approval before Project Approval Committee (PAC).
*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.
*One copy to be submitted to supervisor.

CERTIFICATE

This is to certify that **Annu Arora** is proposing M. Tech. dissertation titled **ENHANCED MULTILEVEL HYBRID ALGORITHM FOR GRAPH PARTITIONING** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation proposal is fit for the submission and the partial fulfilment of the conditions for the award of M. Tech. Computer Science and Engineering.

Date: Signature of Advisor

Karanvir Kaur

UID: 14856

ABSTRACT

Graph partitioning is a traditional problem with many applications and a number of high quality algorithms have been developed. Graph partitioning by multilevel scheme is more popular, which includes three phases coarsening, initial partitioning and uncoarsening. Graph partitioning is also applied on road network graphs, where the main aim is to minimize the cut size. In road network graph, the nodes or vertices can be road junctions, endpoint, etc and the edges are the path, road length. Road networks have various natural cuts as bridges, mountain passes. In this algorithm multilevel scheme of graph partitioning is used and implemented on road network graphs. Recently multilevel hybrid graph partitioning algorithm was proposed using local search refinement strategy and balanced big method for initial partition. In this algorithm simulated annealing is being used as refinement strategy to improve the quality of partition on road network graphs.

ACKNOWLEDGEMENT

I owe a great many thanks to a great many people who helped and supported me during the making of this Dissertation report.

My deepest thank to **Assistant Professor, Ms. Karanvir Kaur**, the mentor of the dissertation, for guiding and correcting me on at every stage of the literature review with attention and care. She has taken pain to go through the topic and make necessary correction as and when needed. She has also supported at each and every step to the finish.

My heartiest thank to **Assistant Professor, Mr. Dalwinder Singh**, Head of Department (Software Engineering, Computer Science and Engineering), for allowing me to take up this topic.

Annu Arora

Master of Technology

(Computer Science and Engineering)

DECLARATION

I hereby declare that the dissertation proposal entitled, ENHANCED MULTILEVEL HYBRID ALGORITHM FOR GRAPH PARTITIONING submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date:

Annu Arora

Investigator

Reg.no-11304935

TABLE OF CONTENT

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Algorithm

Algorithm is a combination of sequence of finite steps to solve a particular problem. Every algorithm takes some data as input and gives some value as output. An algorithm should take zero or more input but should produce at least one output. An algorithm should be deterministic, programming independent and should give correct output. Different data structures are used for storing and managing different types of data. Algorithm can be analyzed by its time and space complexity. Algorithm can be used as a tool for solving a well-defined computational problem. There are different types of problem to be solved. A problem is in P- class if there exist a deterministic algorithm to solve that problem and that problem can be solvable in polynomial time example Binary search, Quick sort etc. A problem is in NP- class if non-deterministic polynomial algorithm is possible for that problem.

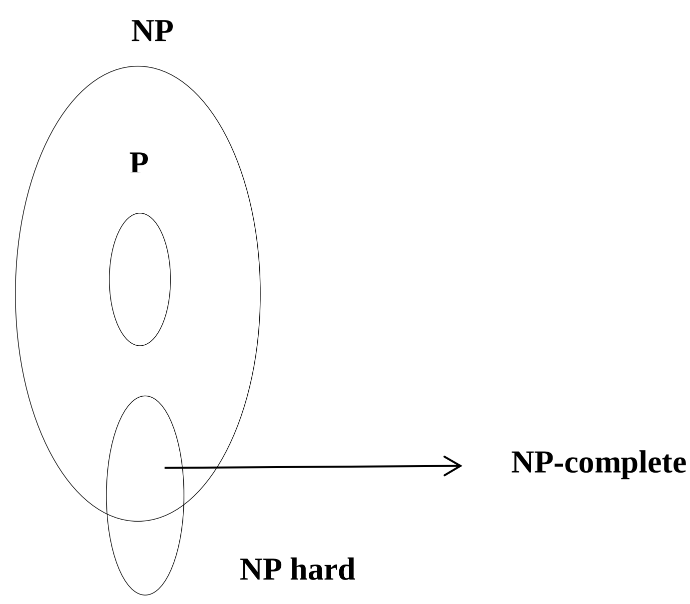

Fig 1.1: NP-complete problems

A problem is said to be in NP-class if it can be proved in polynomial time. A problem which is NP but not P is NP-complete problem. If G is unknown and H is NP-complete problem and if H is polynomially reduced to G then G is said to be NP-hard problem. It is easier to harder conversion.

1.2 Graph Partitioning

1.2.1 Definition

Graph partition is a problem in which a graph G(V, E, We, Wv) is given where,

V= number of vertices,

E= number of edges,

We= weight of the edges,

Wv= weight of the vertices.

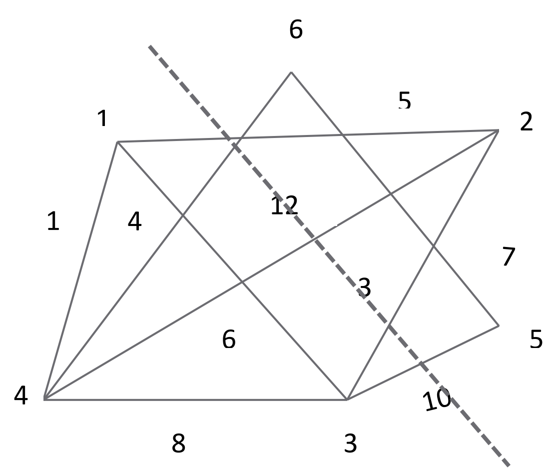

Fig 1.2 Graph for partitioning

This graph is needed to be partition into P1, P2, P3……Pn and choose a partition

P= P1 U P2 P3…..U Pn, such that

a) The number of vertices in which partition P1, P2, P3……Pn is almost same and

b) The number of edges between the partitions should be minimized.

1.2.2 Description

Graph partitioning falls under the category of NP-hard problem. In graph partition problem, a graph G is given in the form of vertices and edges G (V, E) that needs to be partitioned into different smaller graphs. A k-way partition is a technique that divides the graph into k number of components. A partition is better if there is less number of edges cut between the components.

1.2.3 Techniques of graph partitioning

There are several graph partitioning techniques exist. For example Breadth first search (BFS) is the simplest graph partitioning technique, which is used to partition the graph into two components k=2.Geometric partitioning method exploits the network flow technique and identify the cut of the graph. Spectral partitioning method is efficient and most widely used technique. But due to the fact that it consumes more time in computing Eigen values for the adjacency matrix of the graph it is less in use. A multilevel scheme is a new method for graph partitioning which is based on multilevel paradigm. Balanced graph partitioning is the strategy in which the main concern is to partition the graph into partitions of almost same size with less number of edge cut between the partitions. According to it for a parameter v >= 1 no components of the partition should contain more than v.n/k number of vertices. For k=2 and v=1 this problem becomes similar to minimum bisection problem. There are different approaches for graph partitioning as the size of graph grows in size. There is streaming algorithm for graph partitioning, which uses their approach to partition the distributed graph of large size. As the graph size increases it becomes complex to cluster it or to analyze it so it requires to be broken done into components. Multilevel scheme of graph partitioning is considered to be better approach of graph partitioning.

Multilevel graph partitioning scheme

A Multilevel graph partitioning algorithm includes three steps. Initially Coarsening the edges and vertices, then partitioning the graph and finally uncoarsen it with refinement to construct the final partition of the graph. Coarsening phase involves transforming the given graph G into smaller graphs G1, G2, G3…so on these newly formed graphs are taken as input for the next phase i.e. partitioning phase. Partitioning phase is followed by some refinement strategies to get the finest partition. The last phase is uncoarsening of the graphs it is the reverse of the coarsening phase. After this phase we get the final partition of the original graph. For example the three steps in multilevel graph partitioning are shown below:

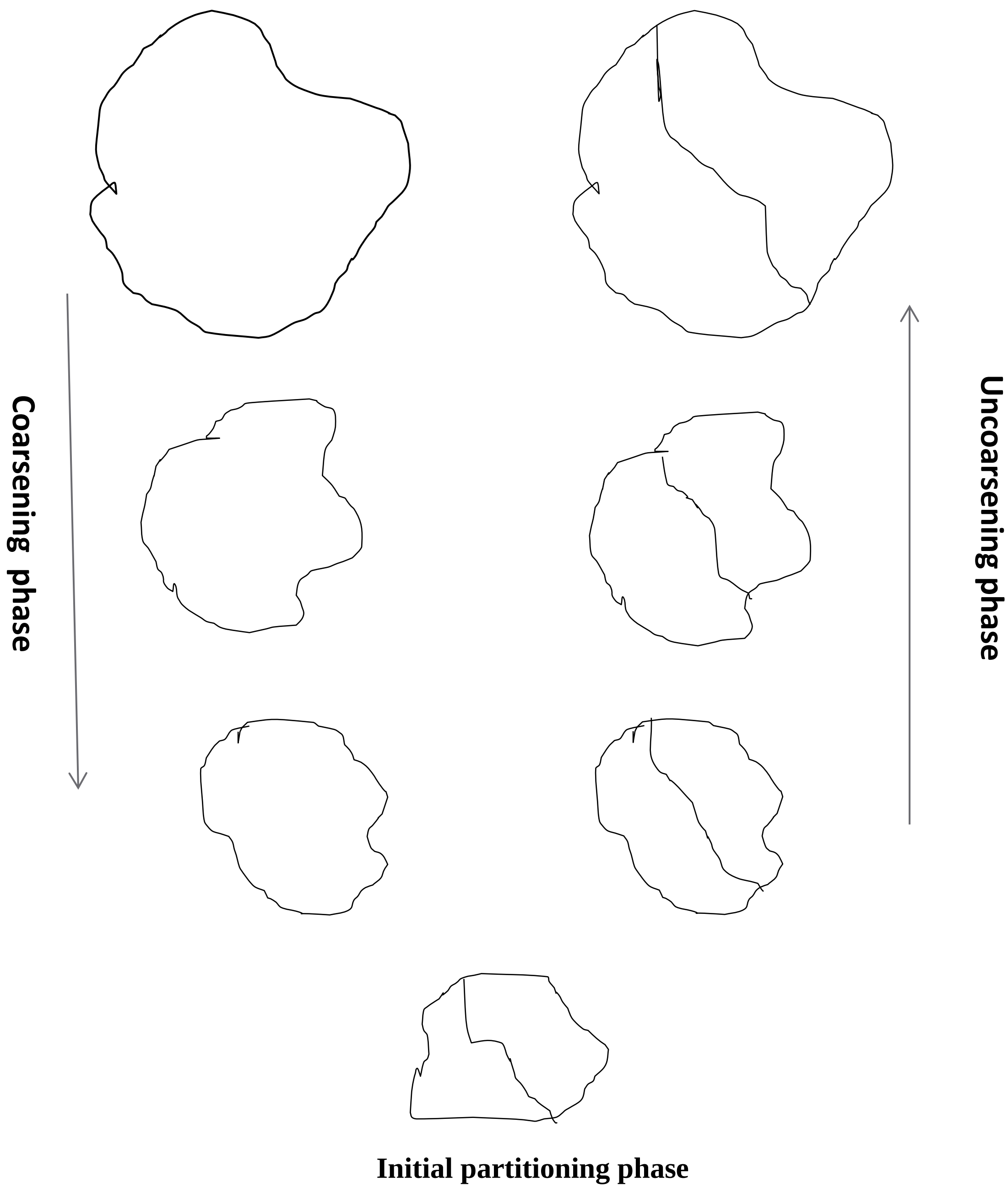

Fig 1.3: Three steps of multilevel graph partitioning

There are different methods for coarsening phase. Two main approaches to coarsen the graph are based on random matching and other is matching heavy edges. To coarsen the graph, first matching is computed which is set of edges that do not incident on the same vertex. Matching should be maximal matching and a matching is said to be maximal if any edge of the graph which is not in the matching set has alteast one of its end point matched.

Random Matching: in random matching algorithm a matching is generated using random algorithm. Vertices are visited randomly. If a vertex u is not matched then we find its adjacent vertex v if it is also unmatched then we choose the edge uv in the matching set. If no adjacent vertex is found unmatched then the vertex u remain unmatched in random matching.

Heavy edge matching: in heavy edge matching, a matching set consists of edges with heavy weight. Their main goal is to minimize the edge cut and also the cut cost. The weight of the edges of the matching set is let W (Mi) and the weight of the edges of the graph is W (E) then the weight of the new coarsen graph will be W (E) – W (Mi). The new coarsen graph does not contain the edges matched in the matching set and the vertices matched are collapsed in the new graph. The vertices which are not matched are kept same in the next graph. Here the heavy edge matching helps to prevent the heavy edge cut in the partitioning.

Light edge matching: Here instead of choosing the heavy weight edges we choose light weight edges. The new coarsen graph has less reduction in the total weight of the graph. Though it does not have much significance but in this case the degree of the new coarsen graph is higher than the original graph. Light weight matching helps in certain partitioning heuristics like KL.

Heavy clique matching: in this method matching is computed by collapsing the vertices whose edge density is high. Here matching set consist of edges whose edge density is maximal. This scheme helps to generate a good bisection of the graph. Randomized algorithm is used to generate the heavy clique matching.

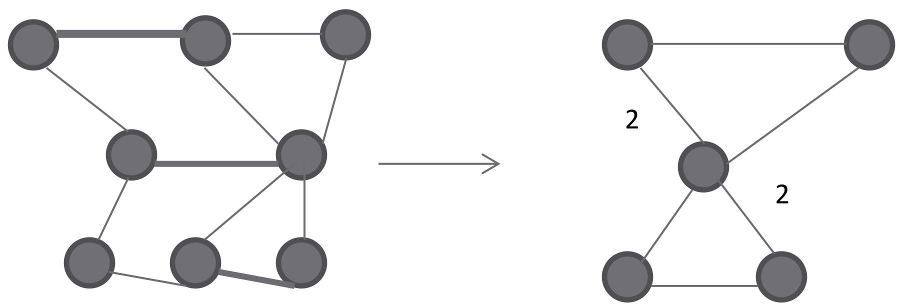

Fig 1.4 Coarsening of graph

There are different algorithms used for partitioning the graph:

Spectral bisection: in this algorithm spectral information is used for graph partitioning. It uses the concept of eigenvector also called as Fiedler vector. Eigen vector is computed using Lanczos algorithm which is an iterative method. This approach cannot be used for large graphs as it consumes much time in calculating the eigen vector.

KL algorithm: KL algorithm is an iterative method, it initially partition the graph into two parts. In each iterative process it searches for the vertices from each part of the graph by swapping the vertices such that the edge cut is reduced. If such partition exits then it becomes the partition for the next iteration. This process continues until two partitions are obtained equal. Each iteration of this algorithm takes O (|E| log |E|) time. Several improvements are done to this algorithm by Fiduccia and Mattheyses which reduces its complexity to O (E) .

Graph growing partitioning algorithm: this algorithm is one the algorithm used for bisecting the graph. It initially chooses one vertex and use BFS to cover the vertices around to create the component of the graph.

Greedy graph growing partitioning algorithm: here in this algorithm, they choose the initial vertex for partitioning such that the edge cut is low. They start by initially choosing four vertices and select the partition for which the edge cut is minimum. It is observed that it takes less time than GGP for partitioning the graph and produces better cut than by GGP.

Balance big method: this method of graph partitioning is based on partitioning the graph into components of almost same size. The difference between number of vertices in the partitions should not be large. This method includes two algorithms in it-

1. Balanced
2. Deterministic Greedy

According to balanced algorithm the vertex with higher degree will be assigned to partition of smaller size. And according to deterministic greedy method vertices with low degree will be assigned to partition of bigger size.

Several techniques are being used for refinement:

KL refinement

Boundary KL refinement

FM refinement

Tabu search

Hill climbing: hill climbing is one of the refinement algorithm used for improving the quality of the partition. Once the initial partition is done on coarsen graph refinement algorithm is applied on it so that the edge cut is reduced. In this algorithm move gain is computed for the vertices present in the partitions. The vertices which are at the boundary of the partition are used for computing move gain. The vertex with highest move gain is choosen from the partition of higher size to move to the partition of smaller in size. Thus in this way the edge cut cost is reduced.

Simulated annealing: Simulated Annealing is the optimization approach based on physical annealing, an enhanced version of local optimization. It works under the random number generator and a control parameter called the temperature, cooling ratio r, temperature length.

There are certain generic parameters used as-

INITPROB: it is used to determine the initial temperature for the set of runs. The value of INITPROB is considered as the starting temperature.

TEMPFACTOR: it is also named as cooling ratio r whose value lies between 0 and 1.

SIZEFACTOR: it is used to set the temperature length L as N*SIZEFACTOR, where N is the size of the expected neighbor.

MINPERCENT: it is used for testing whether annealing run is frozen or not. A counter is maintained which is incremented each time when temperature is reduced and is set to zero when optimum solution is achieved.

1.2.4 Applications of graph partitioning

- Designing telephone network
- In load balancing while minimizing communication
- VLSI layout
- Used in data mining and clustering
- Image segmentation
- Physical mapping of DNA
- Helps in finding communities
- Parallel computing

1.3 Road network graphs

Road network graph is the graph G (V, E) with V vertices E edges where vertices are the end points or the junctions and edges are the path connecting between them. If W (E) is the weight of the edges then in road network graphs the distance between the edges or the time taken to travel from one junction to another can be considered as the weight of the edge.

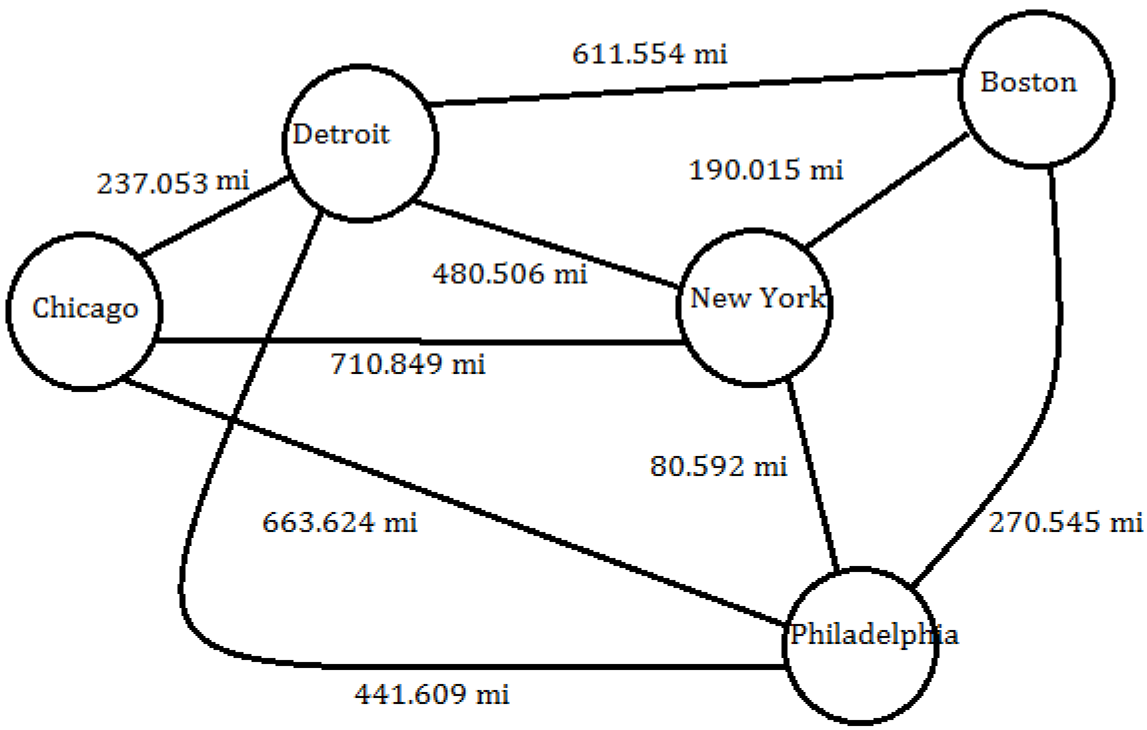

Fig 1.5 Road network graph

In the above figure the vertices are the cities and the edges are the path between them. It is showing the network graph where the edge weights are the distance between the cities. The road network graph is being partitioned using the method of natural cuts in PUNCH algorithm. They defined natural cuts as bridges, mountain passes etc. This algorithm works well for the road network graphs. Road network graphs are used to analyze the network flow or control the network management. As the graph grows in size it becomes complex to analyze it, so the graphs are needed to be divided into subgraphs to simplify the accessing of the graphs. Road network graphs are used to monitor traffic control, planning routes etc.

Chapter 2

REVIEW OF LITERATURE

Dr.S.Padmavathi et al (2014) proposed a multilevel hybrid algorithm for balanced graph partitioning. In this paper a graph is initially partitioned using balanced big method in order to improve the initial solution quality. Further local search refinement procedures are used to improve the quality of partition. They compared the balanced big method for initial partitioning followed by local search (Hill Climbing) refinement with the existing random initial partition with tabu search refinement. They concluded that the proposed approach improves the partition quality compared to the existing method. They implemented their algorithm in Microsoft C#.Net and compiled in Microsoft .Net Framework 4.5.

Somayeh Sobati moghadam (2013) proposed two methods for automatic visualization of the metro map. They used the concept of simulated annealing for changing the position of the node. Their first method produced good quality metro map and second method works well for dense graphs.

Isabelle Stanton et al (2012) proposed streaming algorithm for graph partitioning, they used simple, natural heuristics and compared with hashing and METIS and has found significant improvement. In this paper they mainly focused on balanced graph partitioning using different methods for partitioning, such as Balanced method, Deterministic Greedy method etc. They have used different techniques for their stream order, as BFS, DFS and random for partitioning the graph. They concluded that their heuristic can improve the edge cut in the distributed graph.

Una Benlic et al (2011) used tabu search approach for balanced graph partitioning in multilevel graph partitioning. They included coarsening phase, initial partition, refinement using tabu search and then uncoarsening phase. Their approach of tabu search has resulted better than the partitioning schemes such as METIS or CHACO. In this paper they have explained how vertices are moved across the partitions to improve their quality.

Daniel Delling et al (2011) developed new algorithm called PUNCH for partitioning the graph which is based on natural cuts. This algorithm contains two steps, the first step performs minimum cut computation that reduces the graph size and second phase uses greedy and local search method to assemble the final partition. This algorithm performs well on road network graphs. As a conclusion this algorithm PUNCH preserves the original structure of the graph, also reduces its size which helps to run the more time consuming routines used for assembling the graph. They obtained the best partition method for road networks improving the previous bound by 10% on average, but it is slower than previous existing graph partitioning algorithm.

Cedric Chevalier et al (2009) compared two different methods for graph partitioning geometric and algebraic. Graph partitioning is used in applications like VLSI designing, in parallel computing, proper task scheduling, etc. The main purpose of partitioning is to divide the graph into number of components of almost same size, so that the cut count is reduced. The main objective of a multilevel based algorithm is to divide the problem into different levels i.e coarsening that represents the problem but with less degree of freedom. In this paper two coarsening schemes AMG (algebraic multigrid) scheme is compared with heavy edge matching (HEM). As a result of the experiment without the use of any refinement strategy WAG performs better than HEM. The second experiment was performed using FM and hill-climbing. Here also WAG proved better than HEM with smaller standard deviation. Finally based on some more experiments it was observed that HEM was less efficient than WAG. As a result they recommended use of WAG instead of classical HEM. WAG schemes seem more powerful than SAG (strict aggregation) coarsening since they provide good quality results but with poor refinement. The partitioning obtained during their experiments can be improved by using some more refinement strategies.

Robert Krauthgamer et al (2008) proposed an algorithm for k-balanced partitioning problem. Their main goal was to partition the graph into k partitions of almost equal size. The problem is to partition the graph with minimum number of edge cut and balanced cut. In this algorithm they improved their previous work. They used semi defined relaxation which combines l^2_2 metrics with spreading metrices.

Sanjeev Arora et al (2008) surveyed some techniques that results with some progress in the problem of graph partitioning, which is an NP-hard problem. They described two approaches to graph partitioning one was geometrical approach and second approach was routing flows in the graph for which they used traffic flows in a road network. These both approaches are dual views of each other resulting in an algorithm which is fast and produces high quality cuts.

George Karypis et al (1998) presented and studied a graph partitioning algorithm in which the size of the graph is reduced by merging vertices and edges, they found a k-way partition for the original graph in O(|E|) time, which is faster than the previously proposed multilevel recursive bisection algorithm by a factor of O(log k). Their main contribution in this was to find a high quality and less expensive refinement algorithm. There are three stages in this algorithm coarsening, initial partitioning and refinement. During the coarsening phase three methods are used to find the maximal matching for coarsening, heavy edge matching (HEM),random matching (RM)and modified heavy edge matching (HEM*). For initial partitioning they used multilevel recursive bisection algorithm that produces good initial partitioning. In uncoarsening phase local refinement algorithms based on Kernighan-

Lin (KL) partitioning algorithm is used. New refinement algorithm has developed. Their experiments have shown that multilevel k-way partitioning algorithm is significantly faster than recursive bisection based k-way partitioning schemes.

George Karypis et al (1998) presented a new coarsening method named as heavy edge heuristic in the (multilevel scheme of graph partitioning). In the coarsening phase the vertices are collapsed to get a new vertex. In heavy edge method they collapse based on some matching i.e heavy edge matching. They also presented a new version of (KL) algorithm used for refining during the uncoarsening phase. In this paper they have perform different methods for coarsening ,initial partition and refinement schemes and concluded that multilevel partitioning scheme work quite well. The multilevel algorithm that uses HEM and BKL for refinement can be parallelized effectively.

Dimitris Bertslmas et al (1993) have introduced about simulated annealing, its behavior how it works and its applications. Simulated annealing is a probabilistic method for finding global minimum of cost function, it uses temperature parameter and perform its work in iterations and continue until it gets frozen. Temperature is reduced in each iteration. Simulated annealing is also used for optimization procedure. They said that simulated annealing is mostly used in image processing and also mentioned that simulated annealing is used for graph partitioning, graph coloring, travelling salesman problem. It is shown that simulated annealing is proved better than KL algorithm for graph partitioning.

L TAO et al (1992) proposed new algorithm for graph partitioning based on simulated annealing and tabu search and has compared with LPK algorithm. Their experimental study shows that their algorithm produced better solution than LPK algorithm. In this approach they have given the idea of neighborhood structure and believed that the combination of these two algorithms can reduce the running time of the algorithm.

David S. Johnson et al (1989) studied how simulated annealing works as optimization for certain problems. They investigated how it works better than the existing traditional algorithms. All the parameters used for simulated annealing is discussed in detail in this paper. They concluded that simulated annealing is a competitive approach for graph partitioning problem, for random graphs they proved better than KL algorithm.

Chapter 3

PRESENT WORK

3.1 SCOPE OF STUDY

The study undergoes is to develop an algorithm for graph partitioning based on multilevel scheme. As it is concluded from the literature review that though there are more schemes for graph partitioning but multilevel scheme is the best one. But there are different techniques for different stages of this scheme. The study aims to identify and understand what techniques are being used and how they work. Which techniques are better and efficient? The main objective of partitioning the graph is to reduce the edge cut by using proper partitioning algorithm, refinement strategies. The study also identifies how they implemented certain algorithms using the real world graphs.

3.2 PROBLEM FORMULATION AND OBJECTIVES

As we know the graph partitioning problem is NP-hard algorithms are designed to partition the graph in appropriate manner with fine quality. There are several algorithms designed for graph partitioning which are being used in different applications. Road network graphs are the graphs which are also needed to be partitioned. The problem is to generate an algorithm to partition the road network graphs with some more improved refinement strategy that can improve the quality of graph partition.

The objectives are:

1. To propose a new algorithm with better refinement procedure for road network graphs.
2. To reduce the edge cut.

3.3 RESEARCH METHODLOGY

The algorithm will be implemented in MATLAB and real road network graphs will be used for partitioning. The algorithm will work as:

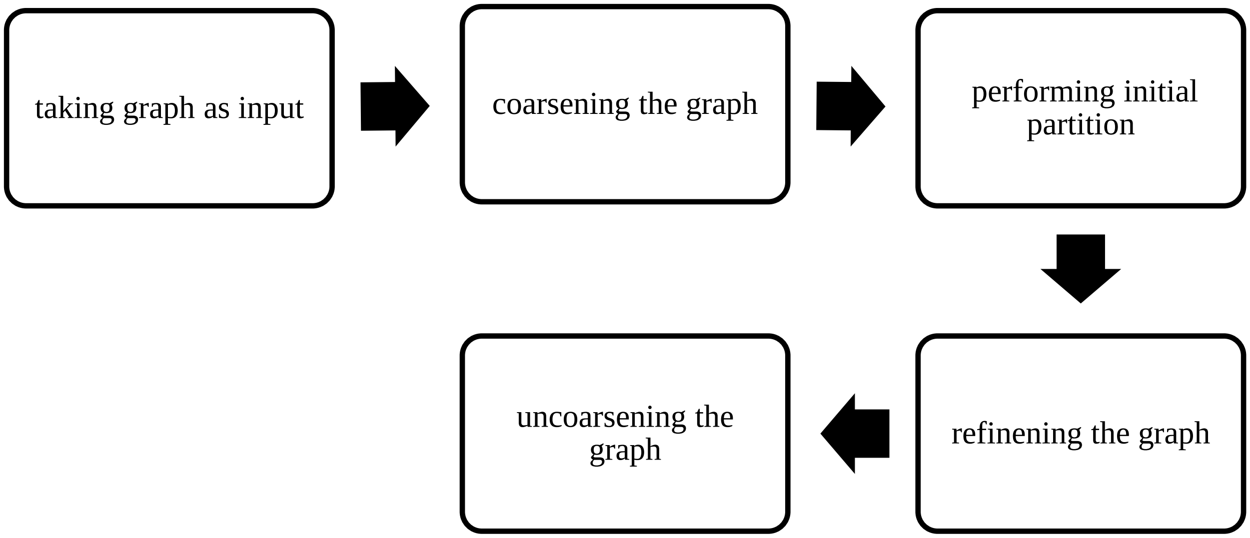

Fig 3.1: Flow chart of the activities to be performed during graph partition.

In the first phase graph G (V, E) will be taken as an input, it will be represented as adjacency list. Next phase is coarsening the graph i.e creating the smaller graph G1, G2..So on from the original graph with reduced number of vertices using graph coarsening method. Then initial partitioning of the coarsened graph is performed. Fourth phase is refining phase which is very important phase to improve the quality of the partition. In the new algorithm we are using simulated annealing for refining the graph and will compare it with other local search refining technique .The last phase is uncoarsening which is the reverse of the coarsening phase the output of this phase is the final partitioned graph. All the phases are discussed in detail below:

I. Coarsening phase

Coarsening phase takes the graph G as input with V vertices and E edges and convert it into smaller graph G_1,G_2,G_3…..so on such that $|V|>|V_1|>|V_2|>|V_3|$ ….and $|E|>|E_1|>|E_2|>|E_3|$.In this algorithm Heavy Edge Matching is used for coarsening the graph. Matching M_i is set of edges such that no two edges have same end points on either side. Heavy edge matching considers the heavy edges in the matching set. We use random algorithm to find this matching by visiting the vertices. For any graph G(V,E) ,$W(e_{ij})$ is the sum of the weight of the edges in G.

M_i = {set of edges}

$w(e_{ij})$ is the sum of weight of the edges belonging to M_i .

The new coarsened graph $G_i(V_i, E_i)$ have edges $E_i = E - M_i$ and

$W_1(e_{ij}) = W(e_{ij}) - w(e_{ij})$

The vertices are collapsed in the new coarsened graph using the matching M_i and the vertices which are not matched are kept as it is in the new graph. Vertices are assigned weight initially and each time when the vertices are collapsed its weight also increases. Vertex weight determines the number of vertices collapsed with it. Heavy Edge Matching is used so that the edges with heavy weight can be avoided from being cut. Coarsening phase is continued until it reaches the coarsening threshold or less than it.Thus in this way coarsening of graph is done to get the final reduced graph.

II. Initial partition

Balanced Big method is used for graph partitioning. After the coarsening phase now the coarsened graph needs to be partitioned such that the size of the graph is balanced. The difference between the size of the partition should not be large. Coarsen degree determines the number of vertices in each partition.

According to balanced big method-

I. The vertices with high degree or weight should be assigned to the partition with smaller size.
II. The vertices with low degree or weight are assigned to partition with bigger size.

We can use BFS, DFS or Random approach for traversing the graph. After partitioning the edge cut cost is computed.

Edge cut cost= $\sum w(e_{ij})$,

Such that e_{ij} connects the vertex v_i belonging to partition P_i and vertex v_j belonging to partition $P_{j.}$

III. Refinement

Refinement is the phase which is used after the initial partition so as to improve the quality of the partition. The main objective of this phase is to reduce the edge cut or the edge cut cost. There are various different techniques for refinement, but in this algorithm Simulated Annealing is used. Simulated Annealing is the optimization approach based on physical annealing, an enhanced version of local optimization. It works under the random number generator and a control parameter called the temperature, cooling ratio r, temperature length L, there are certain generic parameters used as-

INITPROB: it is used to determine the initial temperature for the set of runs. The value of INITPROB is considered as the starting temperature.

TEMPFACTOR: it is also named as cooling ratio r whose value lies between 0 and 1.

SIZEFACTOR: it is used to set the temperature length L as N*SIZEFACTOR, where N is the size of the expected neighbor.

MINPERCENT: it is used for testing whether annealing run is frozen or not. A counter is maintained which is incremented each time when temperature is reduced and is set to zero when optimum solution is achieved.

Simulated Annealing working:

1. Find the initial solution S.
2. Take an initial temperature T >0.
3. Continue until not frozen

3.1 Perform following steps L times

3.1.1 Find a random neighbor S' of S.

3.1.2 Calculate Δ = cost(S') - cost (S).

3.1.3 If Δ<= 0 (downhill move), set S=S'.

3.1.4 If $\Delta > 0$ (uphill move), set S=S' with probability $e^{-\Delta/T}$.

3.2 Set T= rT.

4. Return S.

Here the initial solution S is taken from the output of the initial partition and the final solution is obtained with improved quality.

IV. Uncoarsening phase

The final phase is uncoarsening phase, it is reverse of coarsening. The final partition obtained after refinement is projected back to the previously obtained graphs. Here partition is done on G1, G2, G3 …..and finally on the original graph G.

Algorithm: Multilevel Graph Partition(Graph G)

1. Take the weighted undirected graph as input.
2. While (|V|> coarsening threshold) perform
   Coarsening (G_i to G_{i+1})
3. For all v in V
   If vertex degree (v) is high perform
   Balanced partitioning
   Else
   Perform Greedy partitioning
4. Simulated annealing(initial partition)
5. While (G_i!= G) perform
   Uncoarsening (G_i to G_{i-1})

Chapter 4

RESULTS and DISCUSSION

We have taken15 connected vertices from road network graph of USA from 9th DIMACS implementation challenge. This opened graph is used for partitioning for k=2 and is implemented it in matlab.

For coarsening phase we considered coarsening threshold= 7,in refining phase for simulated annealing, we set its parameters as INITPROB=0.2,TEMPFACTOR=0.5,MINPERCENT=1 and SIZEFACTOR=1. The partition obtained after simulated annealing is compared with hill climbing local search technique and is considered better then it with reduced edge cut cost.

The fifteen vertices are connected as-

Vertex	vertex	Weight
1	2	803
1	5	842
1	6	2428
2	4	591
2	3	617
2	1	803
3	2	617
4	2	591
5	1	842
6	1	2428
6	7	1178
6	8	1095
7	14	1827
7	15	889
7	6	1178
8	6	1095
8	9	1741
8	10	515
9	8	1741
10	8	515
10	11	1611
10	12	335

11	10	1611
12	13	2039
12	10	335
13	12	2039
14	7	1827
15	7	889

Original graph taken as input is shown below:

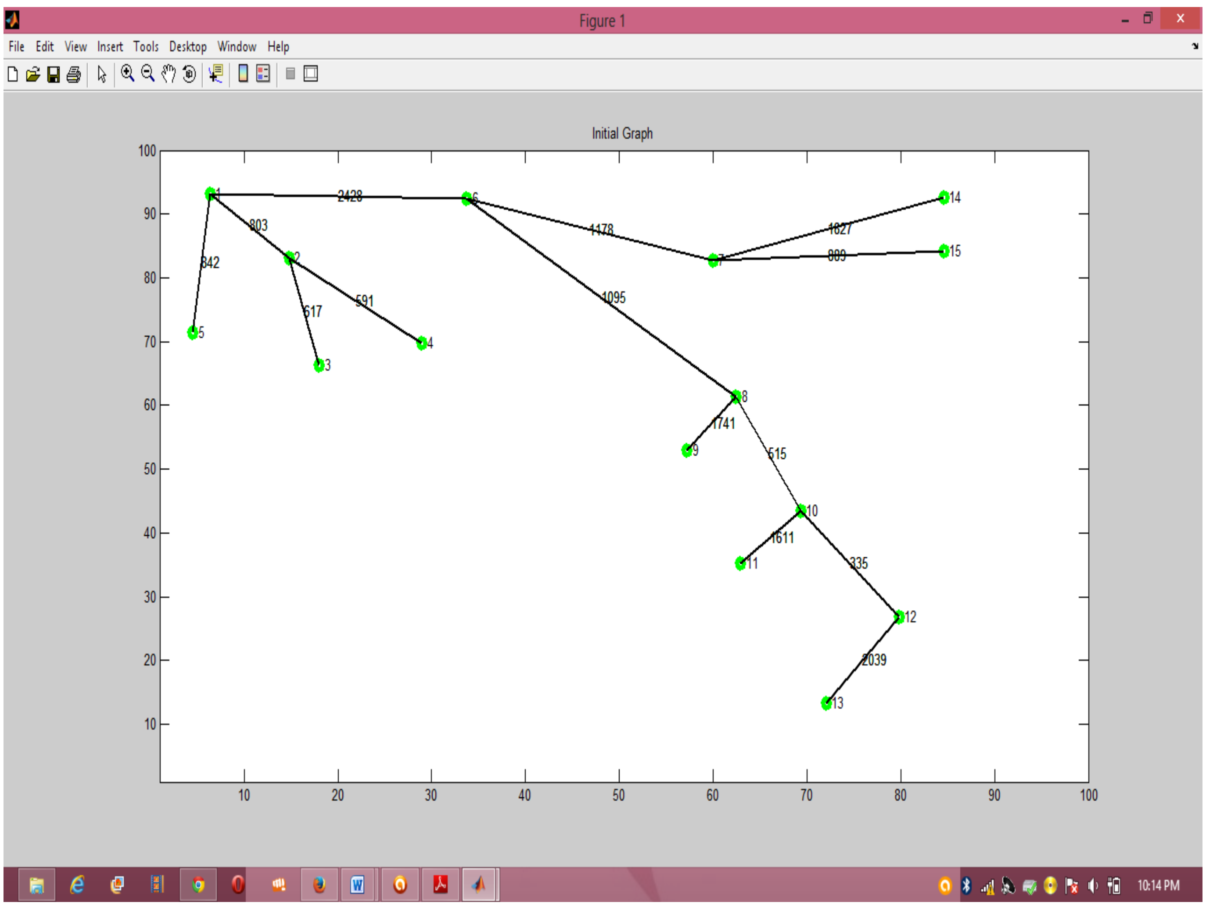

Fig 4.1: Original road network graph

Graph in coarsening phases are shown below:

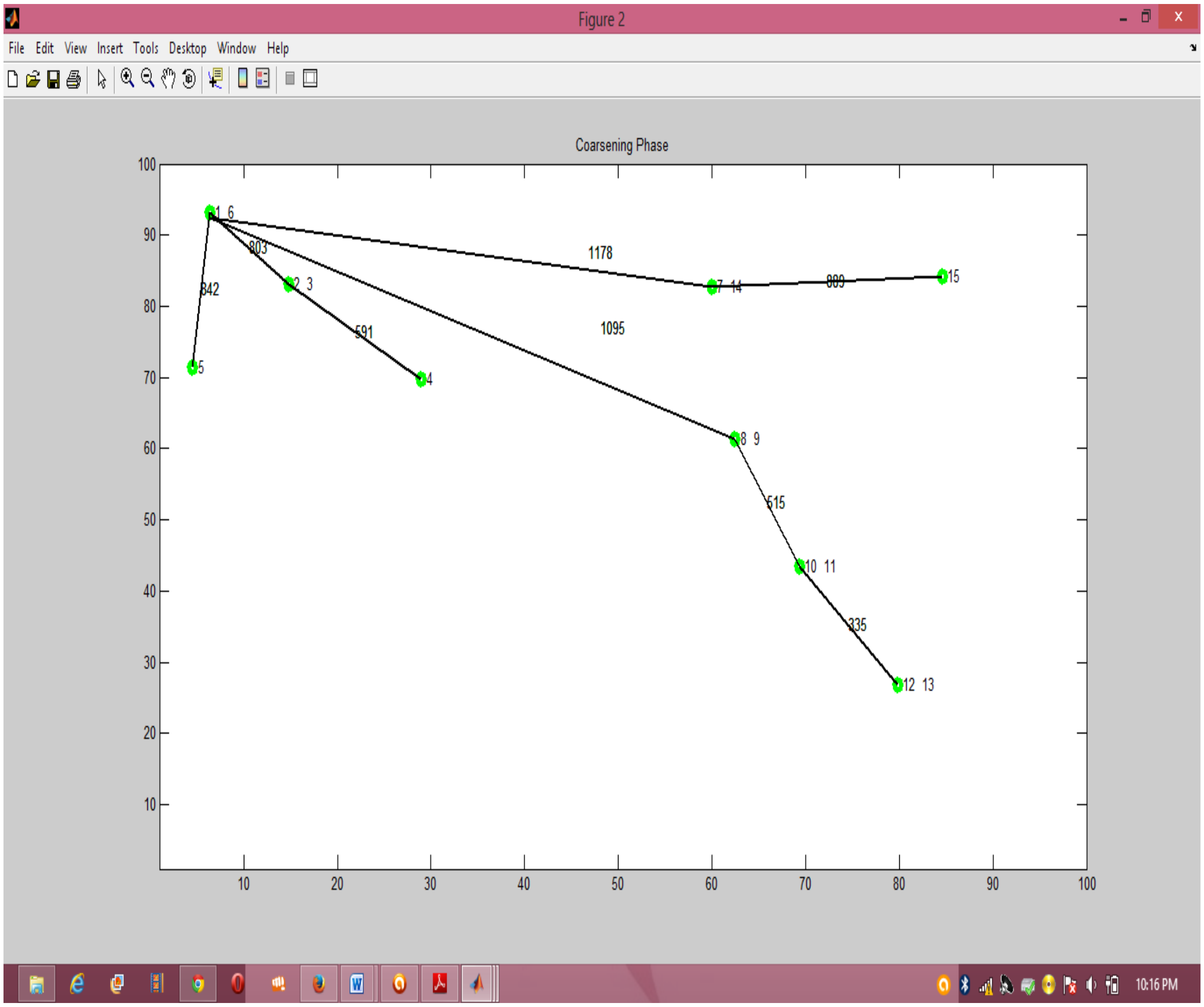

Fig 4.2: Graph during coarsening phase

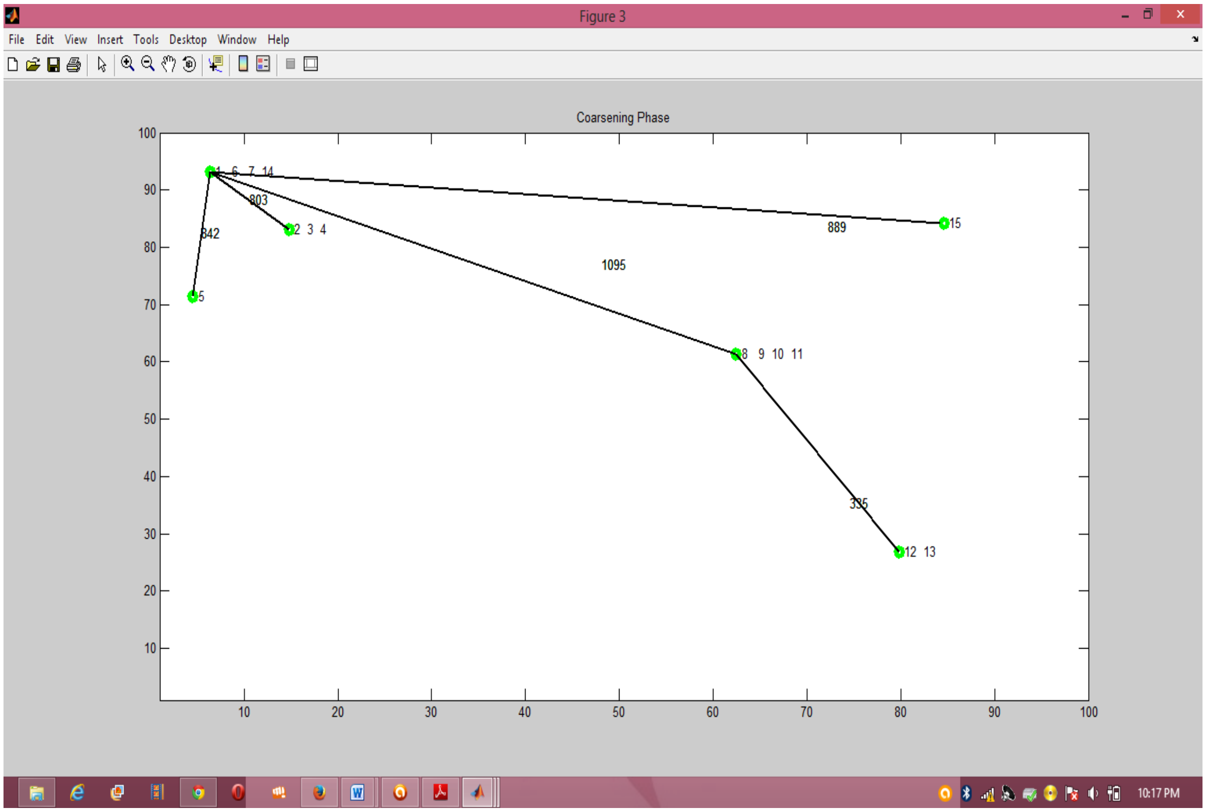

Fig 4.3: Final Coarsened Graph

Graph obtained after initial partitioning is shown below:

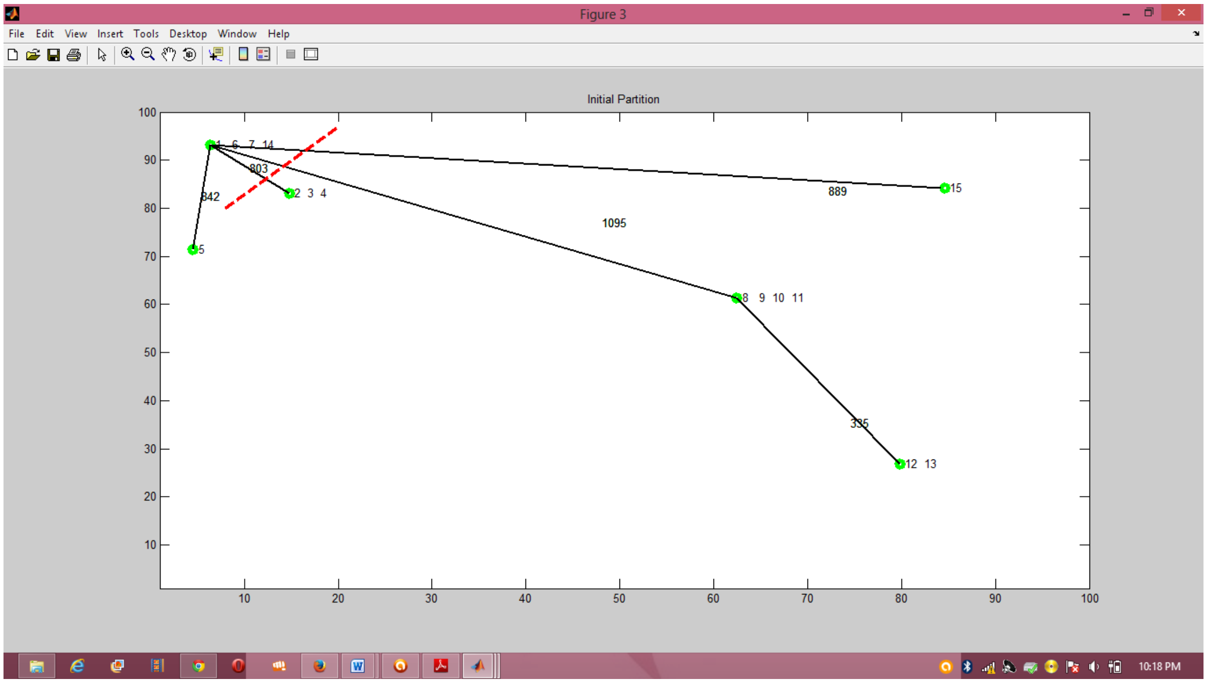

Fig 4.4 Graph after initial partition

Graph after refinement using Simulated Annealing is shown below:

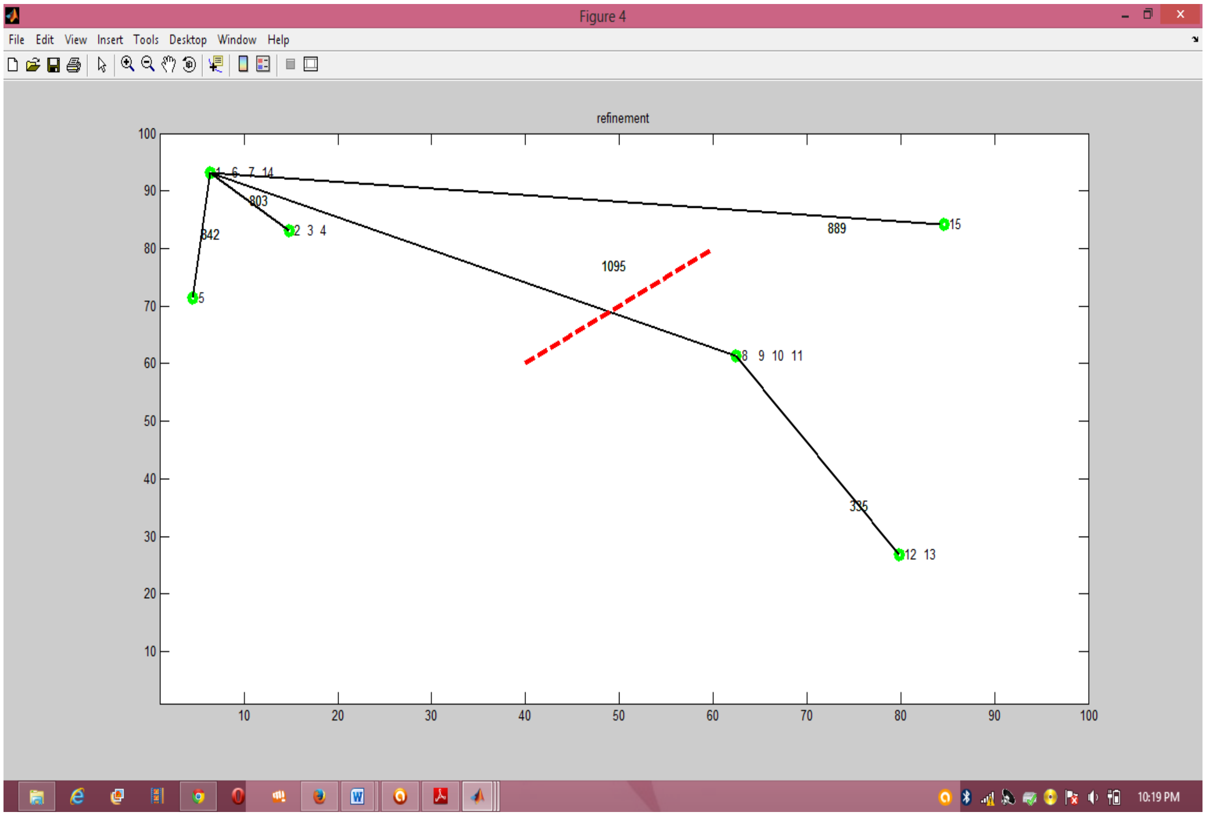

Fig 4.5: Partition after refinement using simulated annealing

Graph obtained using hill climbing is shown below:

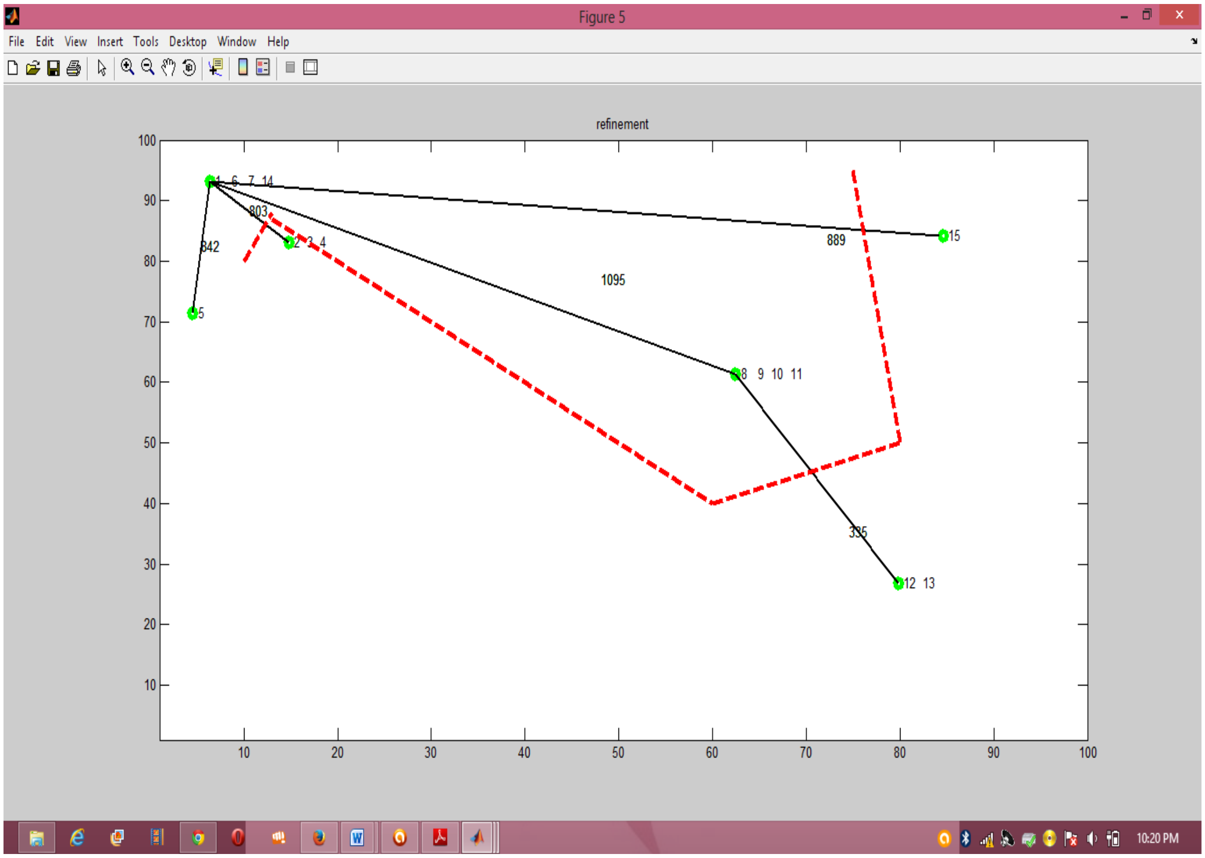

Fig 4.6: Partition after refinement using hill climbing

Graph obtained after uncoarsening phase is shown below:

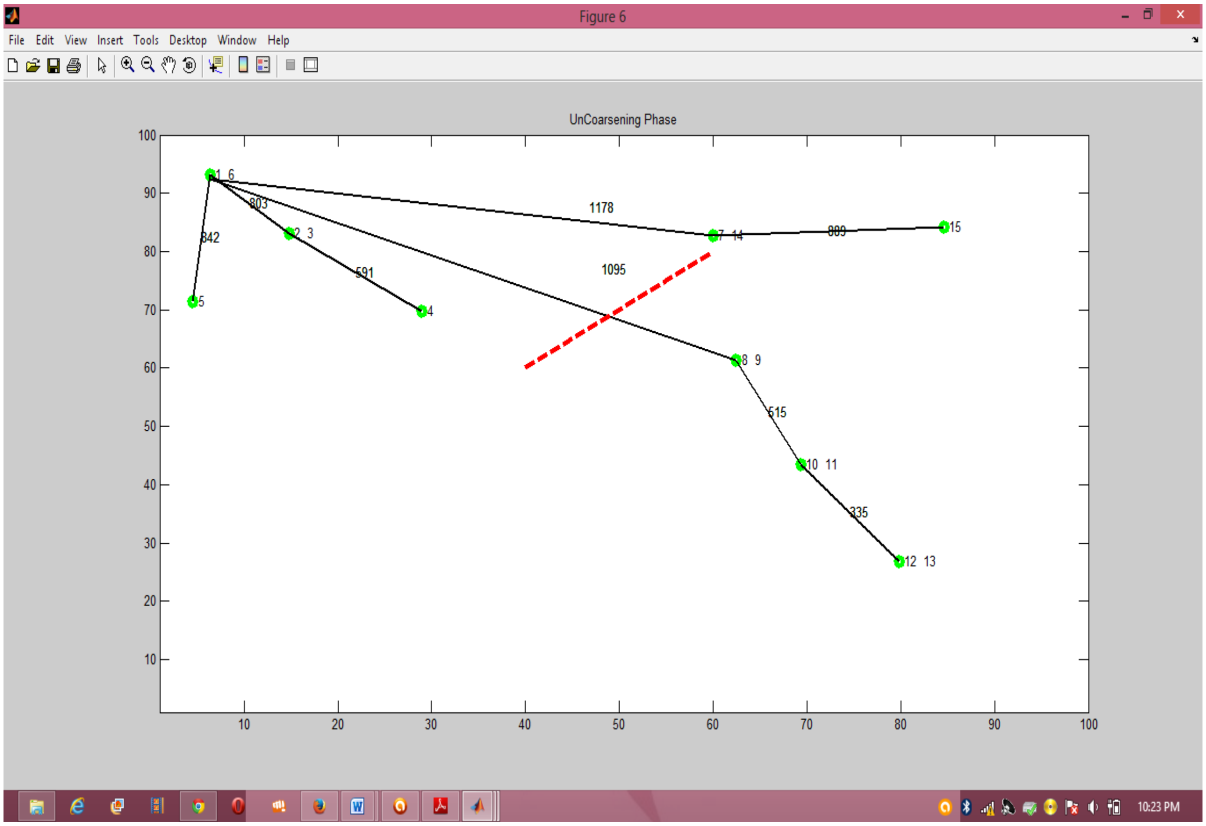

Fig 4.7: Graph during uncoarsening phase

Finally the partitioned on the original graph

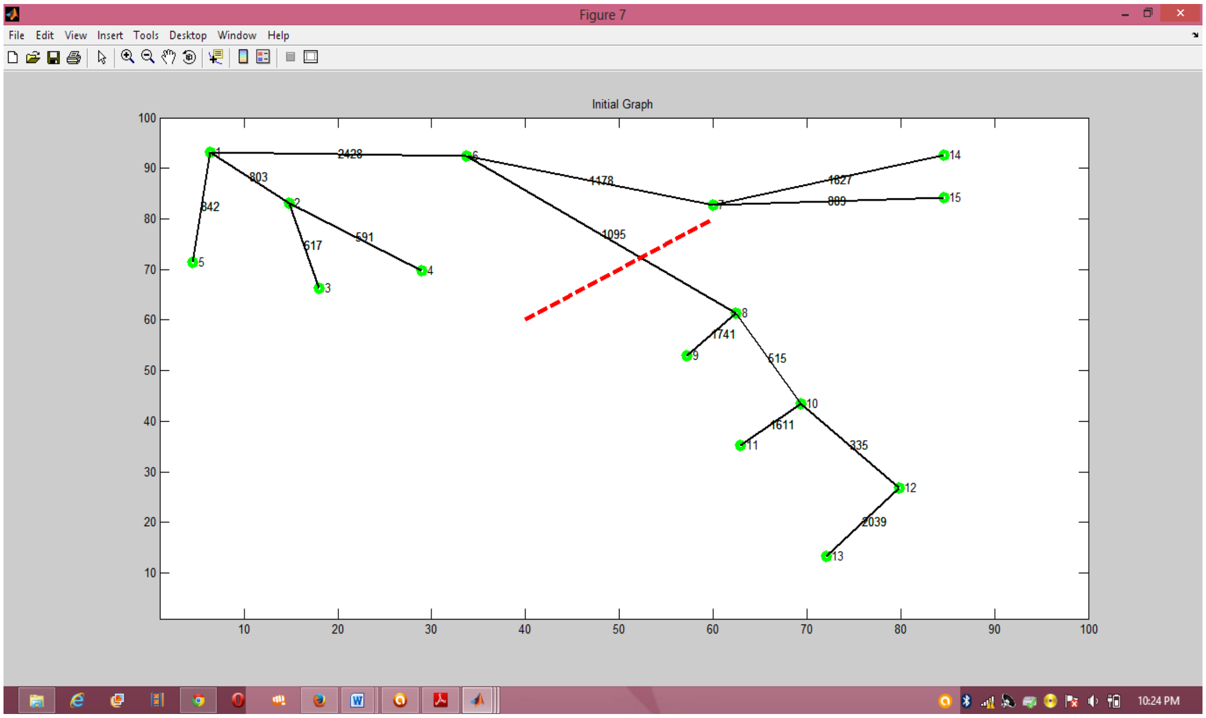

Fig 4.8: Final partitioned Graph

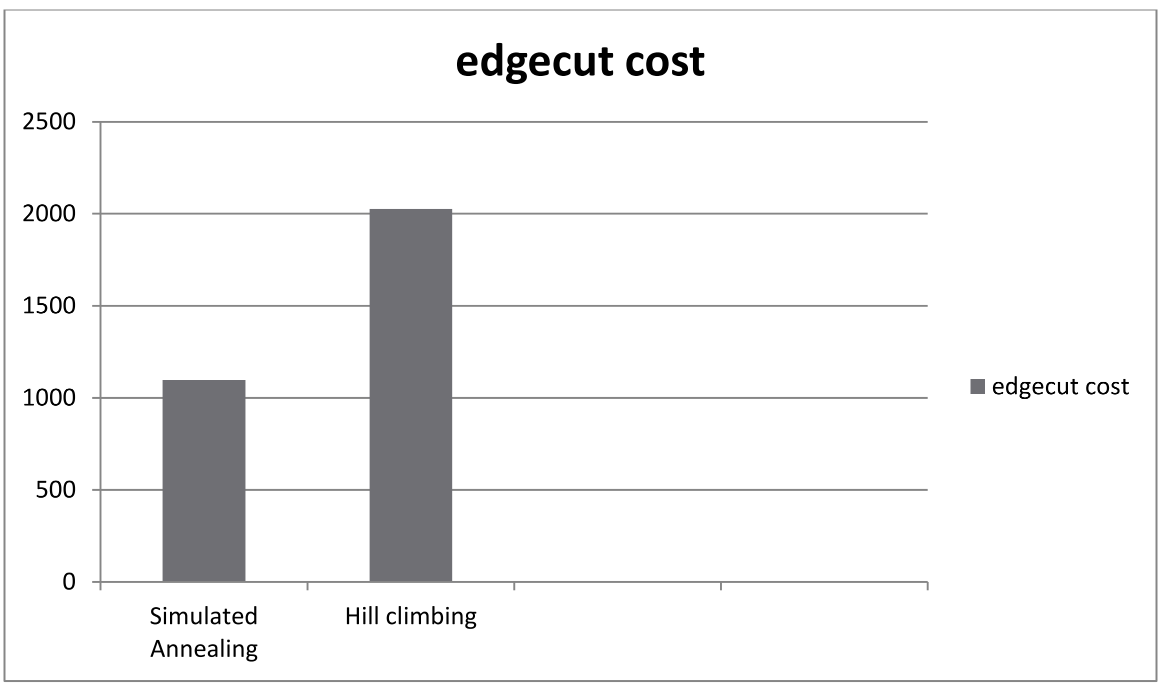

Fig 4.9: Edge cut cost using hill climbing and simulated annealing.

The edge cut cost using Simulated Annealing is 1095; while edge cut cost using Hill Climbing is 2027. So Simulated Annealing has given better result than Hill climbing.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

Graph partitioning is basically an NP-hard problem in which a graph G (V, E) is to be partitioned into different components such that the cut size is minimized, i.e the number of edges linking to vertices of different components should be minimum. Multilevel graph partitioning is the proper approach for partitioning the graph which includes three stages. In our algorithm we have used simulated annealing which works well for partitioning the graph into two components. The edge cut is better and edge cut cost is reduced than the hill climbing refinement algorithm.

5.2 FUTURE SCOPE:

We can use this approach for more number of partitions.

Dynamically any graph can be used for partitioning.

REFERENCES

I. BOOKS

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", Third Edition

II. RESEARCH PAPERS

Cedric Chevalier and Ilya Safro(2009). “Comparison of coarsening schemes for multilevel graph partitioning*”* *Sandia National Laboratories*, Albuquerque, NM, USA, Ccheval@Sandia.Gov. *Argonne National Laboratory*, Argonne, IL, USA, Safro@Mcs.Anl.Gov

Daniel Delling_, Andrew V. Goldberg_, Ilya Razenshteyn_x, and Renato F. Werneck (2011). “Graph Partitioning with Natural Cuts*” Microsoft Research Silicon Valley Mountain* View, CA, 94043, USA.

Dr.S.Padmavathi, Adline George (2014). *“*Multilevel Hybrid Graph Partitioning Algorithm*”, department of* Computer Science and Engineering Thiagarajar College of Engineering Madurai, India

David S. Johnson (1984). “The NP-Completeness Column: An Ongoing Guide” , *Published in J. ALGORITHMS* 5, 433-447 .

David S.Johnson,Cecilia R.Aragon,Lyle A.McGeoch and Catherine schevon (1989). “Optimization By Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning”, Operations Research Society of America ,vol.37,no.6.

Dimitris Bertsimas and John Tsitsiklis (1993),”Simulated Annealing”,Statistical sciences vol. 8,no. 1, 10-15

Emmanouil Spanakis, Christodoulos Efstathiades, George Pallis Marios and D. Dikaiakos (2011). “Real-Time Graph Visualization Tool for Vehicular Ad-Hoc Networks*”, Real-Time Graph Visualization Tool for Vehicular Ad-Hoc Networks* .

Florian Bourse ,Marc Lelarge and Milan Vojnovic(2014),”Balanced Graph Edge Partition”,Microsoft Reasearch.

George Karypis and Vipin Kumar (1998) “Multilevel k-way Partitioning Scheme for Irregular Graphs” , *Journal Of Parallel And Distributed Computing* **48,** 96–129
Article No. Pc971404

George Karypis and Vipin Kumar (1998)"A Fast And High Quality Multilevel Scheme For Partitioning Irregular Graphs", Siam J. Sci. Comput. C° 1998 *Society for Industrial And Applied Mathematics* Vol. 20, No. 1, Pp. 359-392 .

Isabelle Staton, Gabriel Kliot (2012),''Streaming Graph Partitioning for Large Distributed Graphs'',ACM

L.Tao,Y.C.Zhao,K.Thulasiraman and M.N.S Swamy (1992) "Simulated Annealing and Tabu Search Algorithms for Multiway Graph Partition",Journal of circuits,systems and computers

Peter Sanders, Christian Schulz "Engineering Multilevel Graph Partitioning Algorithms", *Karlsruhe Institute of Technology (KIT)*, 76128 Karlsruhe, Germany.

Robert Krauthgamer, Joseph Naor, Roy Schwartz (2008),"Partiotining Graphs into Balanced Components", The Israel Science Foundation

Sanjeev Arora, Satish Rao, and Umesh Vazirani (2008). "Geometry, Flows, and Graph-Partitioning Algorithms", *communications of the acm* | October 2008 | vol. 51 | no. 10

Somayeh Sobati moghadam (2013)," New algorithm for automatic visualization of metro map", International Journal of Computer Science vol. 10.

Scott Kirkpatrick (1984)," Optimization by Simulated Annealing : Quantitative Studies", Journal of Statistical Physics vol 34.

Tarique Anwar, Chengfei Liu, Hai L. Vu and Christopher Leckie (2014). "Spatial Partitioning of Large Urban Road Networks", *Swinburne University of Technology, Melbourne, Australia*

Una Benlic,Jin-Kao Hao (2011) ,"An effective multilevel tabu search approach for balanced graph partitioning"Universite d'Angers, 2 Boulevard Lavoisier,49045 Angers Cedex 01,France Computers and Operations Research

III. WEBSITES

http:// www.cacs.usc.edu
http:// www.msc.anl.gov

APPENDIX

List of Abbreviations

P- Polynomial time

NP- Non Deterministic polynomial time

BFS- Breadth first search

KL- Kernighan- Lin

FM- Fiduccia-Mattheyses

VLSI- Very large scale integration

PUNCH- Partitioning using natural cut heuristics

HEM- Heavy edge matching

LPK- Lee, Park and Kim