COMMUNITY DETECTION & QUANTUM MODELLING

Report submitted for the partial fulfillment of the requirements for the degree of Bachelor of Technology in

Information Technology

Submitted by

Subrata Mondal	-	11700213077
Rudra Pal	-	11700214059
Soumen Mondal	-	11700214068

Under the Guidance of **DR. INDRAJIT PAN**

RCC Institute of Information Technology

Canal South Road, Beliaghata, Kolkata – 700 015

[Affiliated to Maulana Abul Kalam Azad University of Technology, WB]

Acknowledgement

We would like to express our sincere gratitude to DR. INDRAJIT PAN of the department of Information Technology, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he / she took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staff for the gracious hospitality they offered us.

Place: RCCIIT, Kolkata

Date :

………………………………

………………………………

………………………………

Department of Information Technology

RCCIIT, Beliaghata,

Kolkata – 700 015,

West Bengal, India

Approval

This is to certify that the project report entitled "Community Detection & Quantum Modelling" prepared under my supervision by Subrata Mondal - 11700213077, Rudra Pal - 11700214059 , Soumen Mondal – 11700214068 be accepted in partial fulfillment for the degree of Bachelor of Technology in Information Technology.

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which It has been submitted.

………………………………………..	…………………………………………
Dr. Abhijit Das, HOD	Dr. Indrajit Pan, Assosiate Professor

INDEX

Contents	Page Numbers
1. Introduction	6-7
2. Problem Definition	7-8
3. Literature Survey	8-13
4. SRS(Software Requirements Specifications)	13-14
5. Planning	14-15
6. Design	16
7. Results and Discussion	17
8. Conclusion and Future Scope	17-18
9. Reference/Bibliography	18
10. Appendex	19-22

List of Figures

There are 5 figures in our project report. Numbers of figures with Specification are given below:

FIGURE NO. 01 : Clique Percolation Method

FIGURE NO. 02 : Overlapping Community

FIGURE NO. 03 : Betweenness analysis

FIGURE NO. 04 : Link partitioning

FIGURE NO. 05 : Flow Chart

1. INTRODUCTION

Networks are a natural representation for various kinds of complex system, in society, biology, and other fields. One of the most interesting properties of many types of network is their community structure: the existence of groups, or communities, of vertices that are more densely connected to each other than to vertices in other communities. Communities often represent related groups of individuals in the real world. The automatic discovery of network communities is very useful because, for example, it can help throw light on the structure of networks which are far too large for humans to make sense of manually, even with the help of visualization techniques.

Study of this complex networks are the key task for its next level betterment in terms of service, quality and offerings for advanced applications. Being inspired from the colloquial behavior of human societies and communities, Scientists have provided a deep insight for these complex networks and found community structures as key tool for strategic analysis. A complex network is said to have many community structures if that can be separated in different distinct group of members. These groups are identifiable through the behavior of their members. Members of a group seem to have a dense connection or interaction with other members of that group. Incidentally they maintain a sparse communication with the members belonging to the same network but not to a same group.

In this report makes several unique contributions to the state-of-the-art in community detection. These include

(i) analyzing the real-world community structure and observing that the disjoint communities are enough to be processed for discovering overlapping community structure.

(ii)community structure in a network is identified through strength of connections among different members with Betweenness Study .

(iii) detecting overlapping community structures in such large networks.

Nowadays researchers are working on different aspects of community structures within a network perspective. This article reports a novel graph-theoretic approach for detecting overlapping community structures in such large networks. Members of such large networks share high degree of association. This makes some entities to be a part of multiple groups or community structures and the scenario is known as overlapping community. In literature not much of works are reported on this. In following section, some of the recent research trends will be discuss to present the current status of research in this domain.Nowadays researchers are working on different aspects of community structures within a network perspective. This article reports a novel graph-theoretic approach for detecting overlapping community structures in such large networks. Members of such large networks share high degree of association. This makes some entities to be a part of multiple groups or community structures and the scenario is known as overlapping community. In literature not much of works are reported on this. In following section, some of the recent research trends will be discuss to present the current status of research in this domain.

2. PROBLEM DEFINITION

Many algorithms have been designed to discover community structure in networks. Most of these detect disjoint communities, while a few can find communities that overlap. We propose a new, two-phase, method of detecting overlapping communities. In the first phase, a network is transformed to a new one by splitting vertices, using the idea of *split betweenness*; in the second phase, the transformed network is processed by a disjoint community detection algorithm. This approach has the potential to convert any disjoint community detection algorithm into an overlapping community detection algorithm. Our experiments, using several “disjoint” algorithms, demonstrate that the method works, producing solutions, and execution times, that are often better than those produced by specialized “overlapping” algorithms.Our final objective is to find out overlapping communities by analysing the communities.

For example, through service is the principal function of arterials in urban traffic network. When all links in traffic network at the same level of congestion, compared to collectors and local streets, arterials have priority on strengthen the through capacity by coordination of signals to provide for continuous progressive movement at appropriate speeds. Hence, network zoning cannot ignore the priorities of links based on their functions. Disappointing, previous works neglect this characteristic of traffic network. Two characteristics of network are obvious: Firstly, the intersections on roads in high through priority (e.g., Arterials) are sparser than roads in low through priority (e.g., Local Streets). Secondly, the volume of traffic flow and number of lanes on links in high through priority are more than the ones in low through priority. These characteristics make finding one set of reasonable weights for the balance between influences about two characteristics of network becoming a hard work. Hence, previous modeling always leads to the results that arterials in network are zoned into several sub-networks. we focus on three important aspects:

1) characterizing a community to mine the ultimate cause of the formation of the community,
2) discriminating communities to extract the unique features of one community,
3) mining the evolution of a community to uncover its associated history.

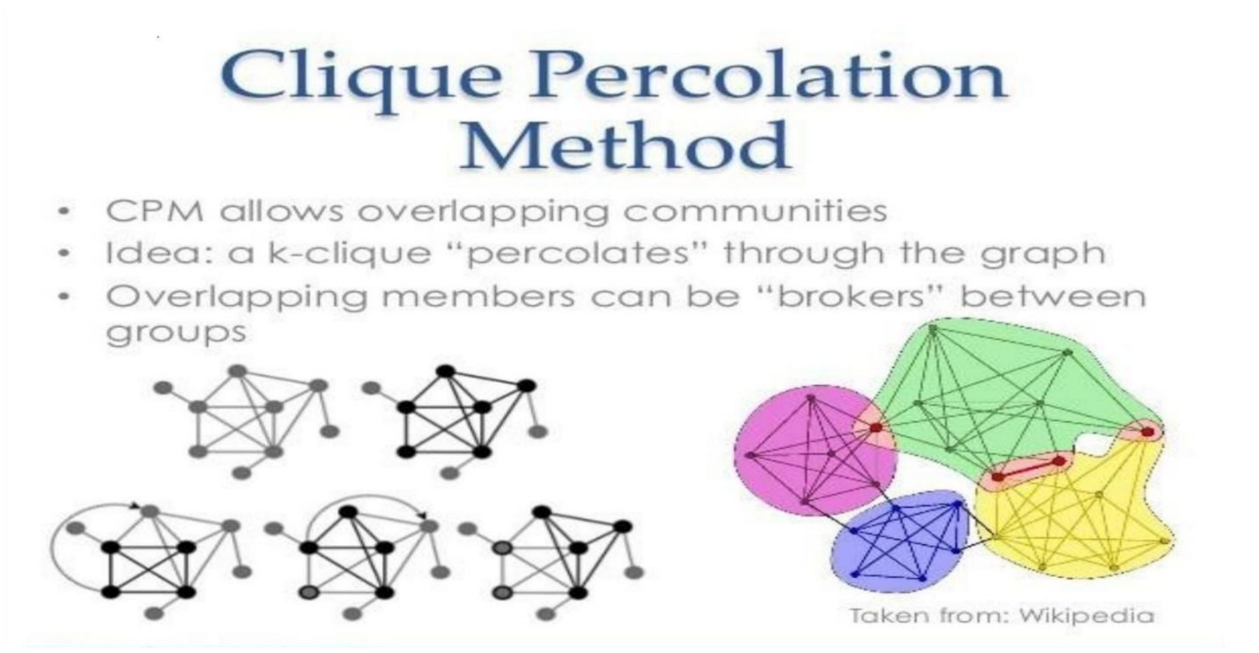
In this paper, except for the definition of resume mining,we also proposed an approach to describe a community by attributes of vertices and topological information and a better method to determine whether two communities are successive by core members instead of the proportion of overlap.

3. LITERATURE SURVEY

Community structure in a network is identified through strength of connections among different members. It requires an analysis on density of connections among these members. Sometimes it evolves as a ratio of dense connection to sparse connection. There are many techniques for measuring this strength of connection among the members. Betweenness analysis is one such competent approach among them. Betweenness centrality is a linear analysis of finding similarity among members of a network. Very few works are reported in literature on overlapping community detection and it is still an open problem to address. In this work a network will be first analyzed for disjoint communities. A novel graph-theoretic analysis will be applied on these disjoint communities to retrieve overlapping communities from that existing network. These disjoint communities will be analyzed for betweenness study. The work will be further extended to form overlapping community structure over the given network.

3.1 Clique Partitioning

A clique partitioning approach towards community detection. The approach is termed as clique percolation method. This method suffers a basic inability to categorize a member node into any community structure if that node doesn’t belong to any clique. Hence the approach is not suitable for large network or any network containing isolated component.


Clique Percolation Method

- CPM allows overlapping communities
- Idea: a k-clique "percolates" through the graph
- Overlapping members can be "brokers" between groups

Taken from: Wikipedia

Fig.1 (Clique Percolation Method)

This approach is reported to retrieve overlapping community structures from a given network.The Clique Percolation Method (CPM) is based on the assumption that a community consists of overlapping sets of fully connected subgraphs and detects communities by searching for adjacent cliques. It begins by identifying all cliques of size k in a network. Once these have been identified, a new graph is constructed such that each vertex represents one of these k-cliques. Two nodes are connected if the k-cliques that represent them share k−1members. Connected components in the new graph identify which cliques compose the communities. Since a vertex can be in multiple k-cliques simultaneously ,overlap between communities is possible. CPM is suitable for networks with dense connected parts. However, it also fails to terminate in many large social networks. CPM introduces a subgraph intensity threshold for weighted networks. Only k-cliques with intensity larger than a fixed threshold are included into a community. In the second phase, the k-community is detected by finding the connected components in the (k− 1)-clique projection of the bipartite representation, in which one type of node represents a k clique and the other denotes a (k−1)-clique. Since each k-clique is processed exactly twice, the running time grows linearly as a function of the number of cliques. SCP allows multiple weight thresholds in a single run and is faster than CPM. Despite their conceptual simplicity, one may argue that CPM-like algorithms are more like pattern matching rather than finding communities since they aim to find specific, localized structure in a network.

3.2 Adaptive Approximation

This approach mainly focuses on betweenness and modularity issue. Betweenness of vertices are found to be processed through modularity analysis. Modularity analysis finally returns community structures. This method yields community structure on static network but inadequate to retrieve overlapping scenarios within the members.In contrast to other methods like fast multipole, panel clustering, etc., the low-rank approximant resulting from the adaptive approximation is not generated by replacing the kernel function of the integral operator. The algorithm uses few of the original matrix entries to compute the low-rank matrix. Note that this does not require to build the whole matrix beforehand. The algorithm will specify which entries have to be computed. The singular value decomposition would find the lowest rank that is required for a given accuracy. However, its computational complexity makes it unattractive for large-scale computations. ACA can be regarded as an efficient replacement which is tailored to asymptotically smooth kernels. Note that not the kernel function itself but only the information that the kernel is in this class of functions is required. This enables the design of a black-box algorithm for discrete integral operators with asymptotically smooth kernels.

3.3 Overlapping Community Detection

With the recent increasing popularity of online social networks services like Facebook , Orkut, Twitter ,Google plus ,studies of community structure are becoming more and more important . community structure is a signature of a complex network. A community is a group of individuals in a social networks. social networking analysis represents the complex network as a graph G formed by two components nodes (V) and edges (E). The V representing the individuals while each edge stands for interaction between nodes in Overlapping communities a node can be member of more than one community.

We first consider the definition of overlapping communities. We formulate minimal properties (axioms) for a set of members to qualify as a community. These are minimal requirements that often appear in the definitions in current use. We attempt to give only the minimal requirements which preserves flexibility and generality. The starting point is a density measure defined on subsets of the vertices. Typically, the density function would represent the communication intensity in the network. The minimality of the requirements outlined by the axioms may lead to implementation difficulties when the number of all sets satisfying the axioms is too big. Because of this possibility, we acknowledge that depending on the specific application, filtering out of some candidate sets based on auxiliary constraints might be needed.

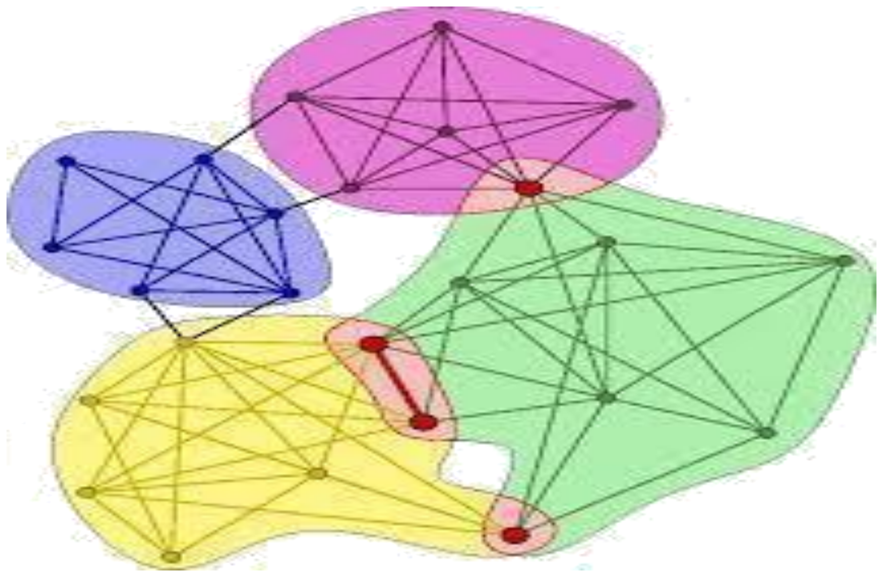


Fig.2 (Overlapping Community)

3.4 Clustering Metrics

Community structure in a network is identified through strength of connections among different members. It requires an analysis on density of connections among these members. Sometimes it evolves as a ratio of dense connection to sparse connection. There are many techniques for measuring this strength of connection among the members. Betweenness analysis is one such competent approach among them which was proposed by Freeman . Vertex betweenness and Edge betweenness are effective metric for measuring communities in a network . Vertex betweenness is a measurement of a vertex (v) in a network (N) with respect to a pair of vertices v1 and v2. A count (ct) of shortest paths between v1 and v2 and a count (cp) of those which pass through v is taken. Then the vertex betweenness of v with respect to v1 and v2 is represented by a ratio between cp and ct. Vertex betweenness of v with respect to whole graph is determined by sum of all such ratios for every possible pair of vertices present in the graph, excluding v. Edge betweenness concept was proposed on vertex betweenness. An edge betweenness measure for an edge (e) is taken with respect to a pair of vertices v1 and v2. It is the number of shortest paths between v1 and v2, those of which passes through e. If there is more than one shortest path then the weight is distributed proportionately.

3.5 Link Partitioning

The basic idea of link partitioning algorithms is to partition links to discover the communities. Two steps of every link partitioning algorithms are:

Step 1: Construct the Dendrogram.

Step 2: Partition the Dendrogram at some threshold.

A node will be identified as overlapping if the links to the node are present in more than one cluster. Links are partitioned by hierarchical clustering in on the basis of edge similarity.

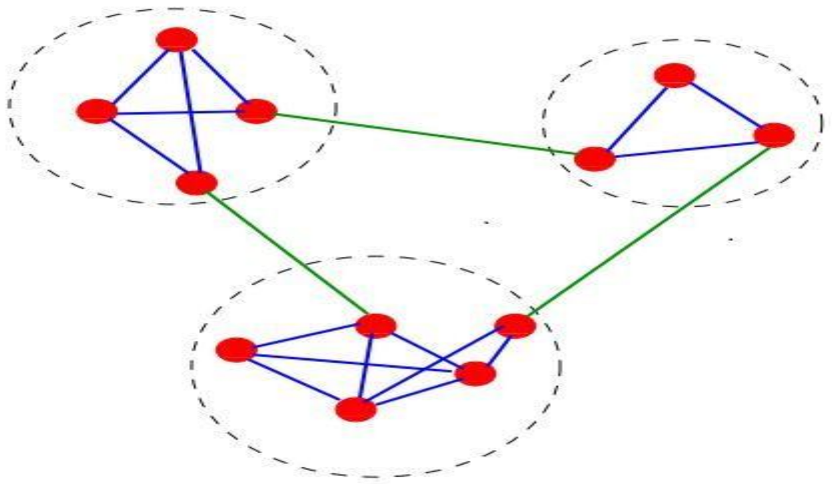


Fig.3 (Link Partitioning)

3.6 Betweenness analysis

In this chapter,we define the graph-theoretic concept of betweenness centrality, which is central to this thesis. This concept takes in to account the global as well as the local feature so features. We present many applications of betweenness centrality and describe the two algorithms ,one for computing the vertex betweenness centrality and the other for computing edge betweenness centrality,for all the vertices and edge sin the graph.Measuring communities in a network there are two effective metric

1) Vertex Betweenness
2) Edge Betweenness

3.6.1 Vertex Betweenness

Vertex betweenness is a measurement of a vertex (v) in a network (N) with respect to a pair of vertices v1 and v2. A count (ct) of shortest paths between v1 and v2 and a count (cp) of those which pass through v is taken. Then the vertex betweenness of v with respect to v1 and v2 is represented by a ratio between cp and ct. Vertex betweenness of v with respect to whole graph is determined by sum of all such ratios for every possible pair of vertices present in the graph, excluding v.

3.6.2 Edge Betweenness

Edge betweenness concept was proposed on vertex betweenness. An edge betweenness measure for an edge (e) is taken with respect to a pair of vertices v1 and v2. It is the number of shortest paths between v1 and v2, those of which passes through e. If there is more than one shortest path then the weight is distributed proportionate.

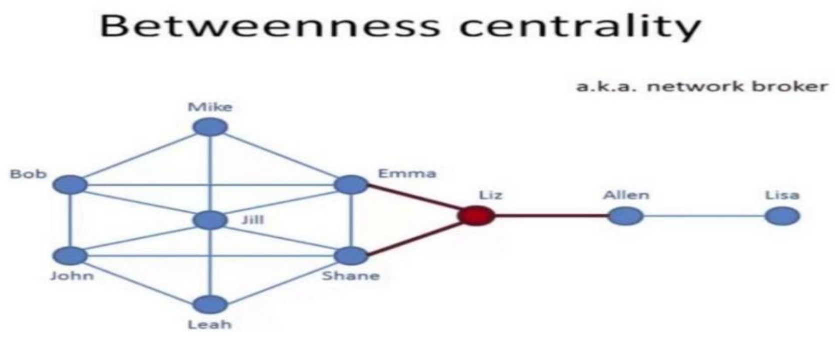


Fig.4 (Betweenness Centrality)

4. <u>SRS (SOFTWARE REQUIREMENT SPECIFICATION)</u>

Proposed algorithm will be implemented in C++ platform. The program was executed in Windows 10 platform, on an AMD A4 microprocessor based chipset board having 4GB primary memory.

The SRS states the functions and capabilities that a software system needs to provide, as well as the constraints that it must respect. The SRS provides the basis for all subsequent project planning, design, coding, and testing. There are many significant benefits to having a SRS document. For starters, the SRS improves communication between your team members by saving and displaying the product feature description in one central location that everybody can easily access. It also prevents confusion within your team by maintaining an up-to-date definition list of all the features included in the project. This way you ensure that everyone develops the same set of features, avoiding a situation in which there are several different versions of product documents out there. And because all that information is available in one document, the SRS makes it easy for new employees to quickly learn the details of the project.

Software Required: - CODE BLOCKS / JOODLE ONLINE COMPILER/ QUINCY

Programming Language : - C++

Version: - 17.12

Size:- 4.10 KB

5. PLANNING

5.1 Proposed Method

Present method applies the concept of vertex betweenness to compute disjoint community structures. These disjoint communities are strictly non-overlapping in nature and there is no single common member among any pair of communities. In the process of forming disjoint communities, each node of the network is analyzed for its vertex betweenness value. Then the nodes are sorted in a descending order of their betweenness values. Nodes or vertices having same betweenness value are grouped together. Each of these groups represents a community within the network and those are non-overlapping in nature. After formation of disjoint communities, these communities are taken in pair for analysis to develop overlapping communities. During vertex betweenness analysis, lengths of shortest path between all pair of vertices are determined. A record of maximum length shortest path among all such pairs is taken in (msp).

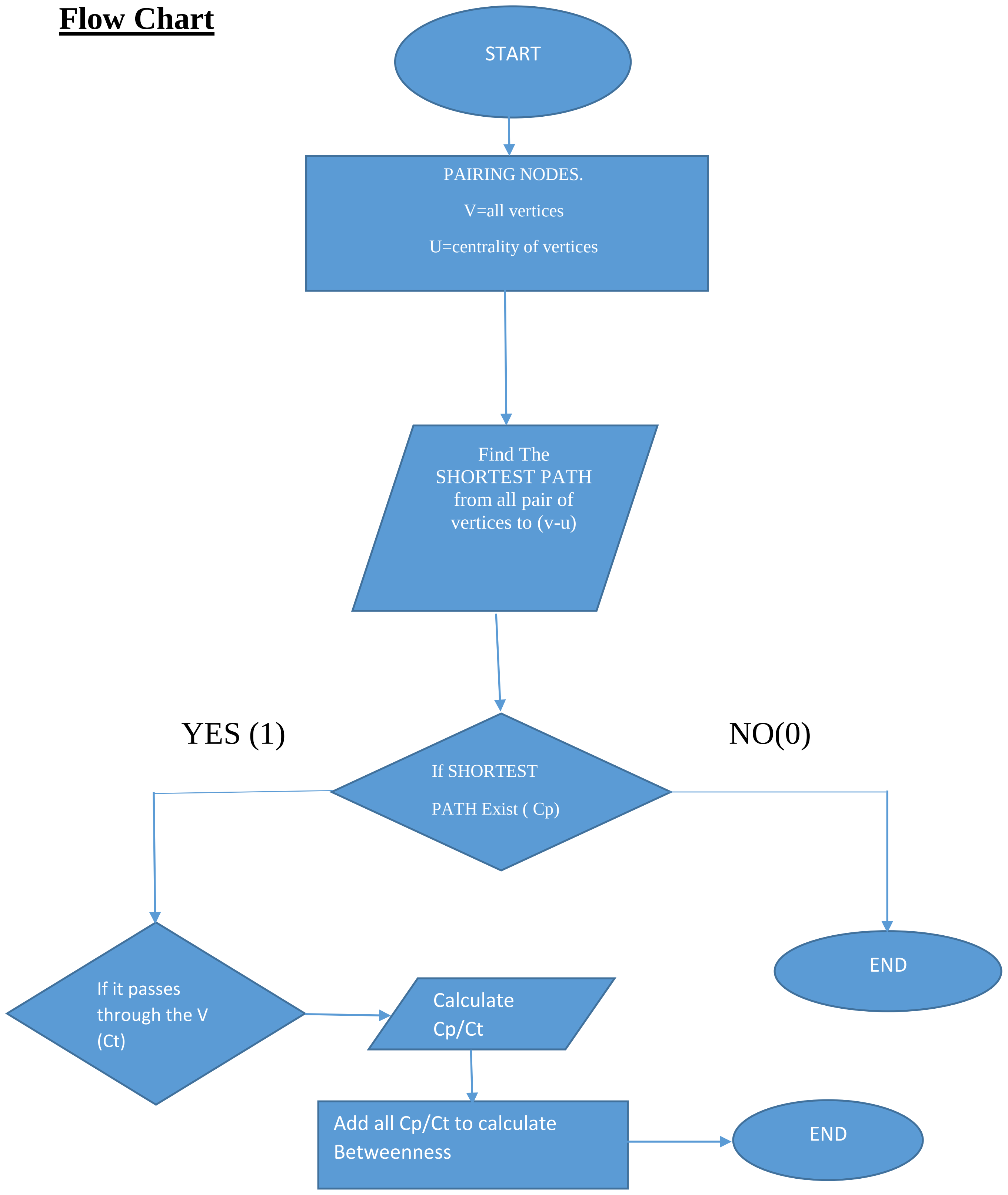
Now communities are taken in pair, where they belong in two disjoint sets. Considering the edges of master network, bipartite connections are introduced between two sets. Now all vertices are individually checked for their incidency (number of edges incident up on that vertex) value in the bipartite graph. If any vertex achieves incidency greater or equal to th then that is merged with another set to form an overlapping community.

5.2 Work Flow Analysis

```
a. read the vertex (V) and edge (E) set of given network
b. for all vertex (v) in V, find out the betweenness (vb)
c. sort all vertices (v) in descending order of betweenness (vb)
d. form non-overlapping clusters (Ck) where k= 1 to m, by
   taking Vi where i= 1 to n from the list S having the same
   betweenness.
e. determine threshold (th)
f. loop: i = 1 to (m-1)
   i. loop: j = (i +1) to m
         1. Consider Ci and Cj in set V1 and V2 respectively
         2. Introduce bipartite connections from (E) between V1
         and V2
            3. Initialize an overlapping community (Cv2) for V2
               a. Cv2 = Cv2V2
               b. v in V1, find incidency(v)
               c. if incidency(v) th
                     i. Cv2 = Cv2 v
                        end if
               d. return Cv2 if Cv2V2
         4. Initialize an overlapping community (Cv1) for V1
               a. Cv1 = Cv1V1
               b. v in V2, find incidency(v)
               e. if incidency(v)
            th Cv1 = Cv1 v
               f. End if
            return Cv1 if Cv1 V1
            end loop; end loop;
```

6. DESIGN

Flow Chart


START
PAIRING NODES.
V=all vertices
U=centrality of vertices
Find The SHORTEST PATH from all pair of vertices to (v-u)
YES (1)
NO(0)
If SHORTEST PATH Exist (Cp)
END
If it passes through the V (Ct)
Calculate Cp/Ct
Add all Cp/Ct to calculate Betweenness
END

7. <u>Experimental Results</u>

We first computed the vertex betweenness distribution for all two networks ,and observed that it follows apower law. We also studied the vertex betweenness. Degree correlation for all two networks. In the edge betweenness distribution for all the three networks ,we saw a strange behaviour,i.e. ,presence of a large fraction of edges with the same betweenness value. To uncover the reason behind this behavior ,we generated random graphs with the same degree distribution as the original networks. We also generated random graphs with different densities and whose degree distribution followed power law with different values of the power law exponent. We plotted the average edge betweenness distribution for all these graphs too. The values of edge betweenness for the edges in a graph we renormalized by dividing it by the total number of edges in the graph. This was done so that we may compare graphs with different sizes, i.e., compare graphs with different number of nodes and edges. This proposed method was primarily simulated on two different customized networks as illustrated below in demo network I and demo network II. These two networks are discussed in the following section called Simulator Network. The purpose of execution on these simulator networks is to judge the effectiveness of the proposed method.

8. <u>CONCLUSION AND FUTURE SCOPE</u>

Overlapping community detection approaches have attracted a lot of attention of researchers in recent years and there is a considerable increase in the number of algorithms published for solving the issue as it has applications in various domains like microbiology, social science and physics. Analyzing community structure in social network has emerged as a topic of growing interest as it shows the interplay between the structures of the network and its functioning. This paper tries to review all popular algorithms for overlapping community detection with their strengths and weaknesses. We have tried our best to review all popular algorithms, but the study is by no means complete as there are newer algorithms discovered at a fast rate because of the growing interest of researchers in this domain. We demonstrated how the output of an efficient disjoint community detection algorithm discover the overlapping community structure. This work explores overlapping members across the communities in a network and thus exposes the details of overlapping communities. Our methods discussed above work effectively to make the state of community clear. Systematic way a wide range of network community detection methods originating from theoretical computer science, scientific computing, and statistical physics. Our empirical results demonstrate that determining the clustering structure of large networks is surprisingly intricate.

The research work and its encouraging results open several new directions. All these open questions will provide new research opportunities to the research community. Overlapping community detection approaches have attracted a lot of attention of researchers in recent years and there is a considerable increase in the number of algorithms published for solving the issue as it has applications in various domains like microbiology, social science and physics. We will work further to implement our project in the social network system like facebook and orkut,there are many communities.we will find a way to implement our project to detect the overlapping in those community.

9. **BIBLIOGRAPHY**

1. Basuchowdhuri, P., Chen, J.: Detecting communities using social ties. In: Proc. of IEEE International Conference on Granular Computing, pp. 55 – 60. San Jose, CA, USA (2010).

2. Derenyi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. In: Physics Review Letter, pp.1– 4. (2005).

3. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. In: Proceedings of the National Academy of Sciences of the United States of Ameri-ca, vol. 99 (12), pp. 7821 – 7826. PNAS, USA (2002).

4. Gu, Y., Zhang, B., Zou, G., Huang, M., Jiang, K.: Overlapping community detection in social network based on microblog user model. In: Proceedings of International Conference on Data Science and Advanced Analytics, pp. 333-339. Shanghai, China (2014).

10. APPENDIX

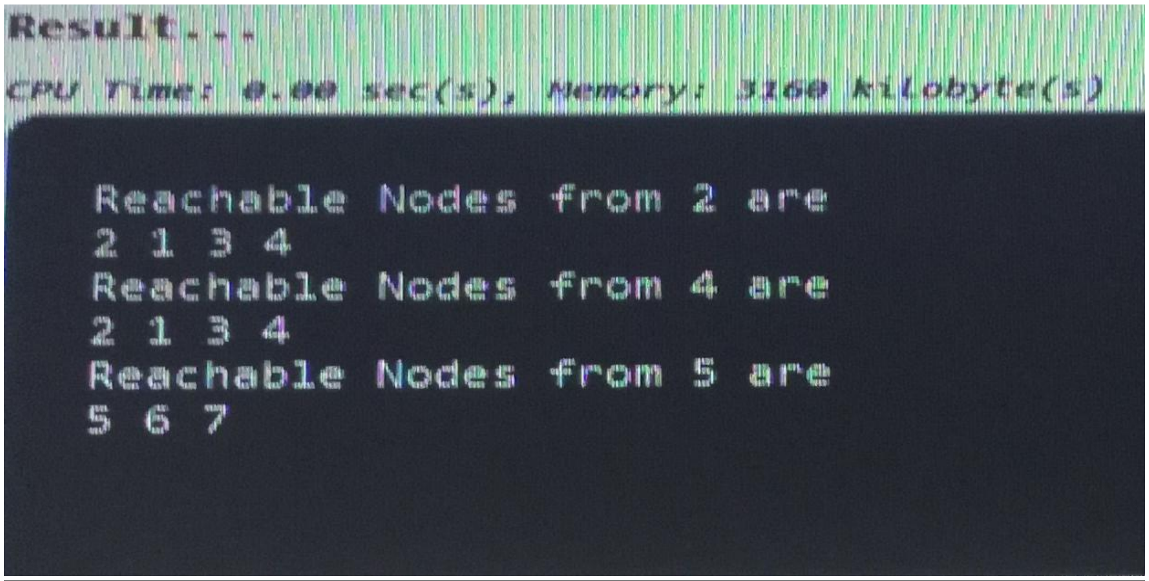
Code

```
#include <bits/stdc++.h>
using namespace std;
class Graph
{
public:
int V;
list<int> *adj;
Graph(int );
void addEdge(int, int);
vector<int> BFS(int, int, int []);
};
Graph::Graph(int V)
{
this->V = V;
adj = new list<int>[V+1];
}
void Graph::addEdge(int u, int v)
{
adj[u].push_back(v);
adj[v].push_back(u);
}
vector<int> Graph::BFS(int componentNum, int src,int visited[])
{
queue<int> queue;
queue.push(src);
visited[src] = componentNum;
```

```
vector<int> reachableNodes;
while(!queue.empty())
{
int u = queue.front();
queue.pop();
reachableNodes.push_back(u);
for (auto itr = adj[u].begin();
itr != adj[u].end(); itr++)
{
if (!visited[*itr])
{
visited[*itr] = componentNum;
queue.push(*itr);
}
}
}
return reachableNodes;
}
void displayReachableNodes(int n,
unordered_map <int, vector<int> > m)
{
vector<int> temp = m[n];
for (int i=0; i<temp.size(); i++)
cout << temp[i] << " ";
cout << endl;
}
void findReachableNodes(Graph g, int arr[], int n)
{
int V = g.V;
int visited[V+1];
```

```
memset(visited, 0, sizeof(visited));
unordered_map <int, vector<int> > m;
int componentNum = 0;
for (int i = 0 ; i < n ; i++)
{
int u = arr[i];
if (!visited[u])
{
componentNum++;
m[visited[u]] = g.BFS(componentNum, u, visited);
}
cout << "Reachable Nodes from " << u <<" are\n";
displayReachableNodes(visited[u], m);
}
}
int main()
{
int V = 7;
Graph g(V);
g.addEdge(1, 2);
g.addEdge(2, 3);
g.addEdge(3, 4);
g.addEdge(3, 1);
g.addEdge(5, 6);
g.addEdge(5, 7);
int arr[] = {2, 4, 5};
int n = sizeof(arr)/sizeof(int);
findReachableNodes(g, arr, n);
return 0;
}
```

<u>Output</u>



```
Result...
CPU Time: 0.00 sec(s), Memory: 3160 kilobyte(s)

Reachable Nodes from 2 are
2 1 3 4
Reachable Nodes from 4 are
2 1 3 4
Reachable Nodes from 5 are
5 6 7
```