2012

Algorithms For Community Identification In Complex Networks

Mahadevan Vasudevan
University of Central Florida

ALGORITHMS FOR COMMUNITY IDENTIFICATION IN COMPLEX NETWORKS

by

MAHADEVAN VASUDEVAN
B.E., Anna University, 2006

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2012

Major Professor: Narsingh Deo

ACKNOWLEDGMENTS

First and foremost, I would like to extend my deepest gratitude to my advisor, Professor Narsingh Deo, for his excellent guidance and encouragement, and also for introducing me to this wonderful science of complex networks. Without his support this dissertation would not have been possible. I would also like to thank the members of my research committee, professors Charles Hughes, Ratan Guha, Mainak Chatterjee and Yue Zhao for their advice and guidance during the entire process.

I am indebted to the faculty and the staff of the Department of Electrical Engineering and Computer Science for providing me the resources and environment to perform this research. I am grateful to my colleagues in the Parallel and Quantum computing lab for the stimulating discussions and support. I would also like to thank Dr. Hemant Balakrishnan and Dr. Sanjeeb Nanda for their valuable suggestions and guidance.

My heartfelt thanks to my parents, Vasudevan and Raji, who have always been supportive of my decisions and encouraged me with their best wishes. I would also like to thank my sister Gomathy, for her words of care and affection during tough times. Special thanks to my friends in Orlando for being there when I needed them.

ABSTRACT

Complex networks such as the Internet, the World Wide Web (WWW), and various social and biological networks, are viewed as large, dynamic, random graphs, with properties significantly different from those of the *Erdös-Rényi* random graphs. In particular, properties such as degree distribution, network distance, transitivity and clustering coefficient of these networks have been empirically shown to diverge from classical random networks.

Existence of *communities* is one such property inherent to these networks. A *community* may informally be defined as a locally-dense *induced subgraph*, of significant size, in a large globally-sparse graph. Recent empirical results reveal communities in networks spanning across different disciplines — physics, statistics, sociology, biology, and linguistics. At least two different questions may be posed on the community structure in large networks: (*i*) Given a network, detect or extract all (i.e., sets of nodes that constitute) communities; and (*ii*) Given a node in the network, identify the best community that the given node belongs to, if there exists one. Several algorithms have been proposed to solve the former problem, known as *Community Discovery*. The latter problem, known as *Community Identification*, has also been studied, but to a much smaller extent. Both these problems have been shown to be NP-complete, and a number of approximate algorithms have been proposed in recent years. A comprehensive taxonomy of the existing community detection algorithms is presented in this work.

Global exploration of these complex networks to pull out communities (community discovery) is time and memory consuming. A more confined approach to mine communities in a given network is investigated in this research. Identifying communities does not require the knowledge of the entire graph. Community identification algorithms exist in the literature, but to

a smaller extent. The dissertation presents a thorough description and analysis of the existing techniques to identify communities in large networks. Also a novel heuristic for identifying the community to which a given seed node belongs using only its neighborhood information is presented. An improved definition of a community based on the average degree of the induced subgraph is discussed thoroughly and it is compared with the various definitions in the literature. Next, a faster and accurate algorithm to identify communities in complex networks based on maximizing the average degree is described. The divisive nature of the algorithm (as against the existing agglomerative methods) efficiently identifies communities in large complex networks. The performance of the algorithm on several synthetic and real-world complex networks has also been thoroughly investigated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Interaction among entities in real-world complex systems have been modeled and studied as *networks* in order to better understand their unique properties. Each *node* of the network corresponds to an individual object of the system and the *edge* symbolizes the interaction between these individual entities. The term *Complex Network* refers to any such network derived from a real-world complex system. Empirical studies show that most of the real-world networks such as the Internet, the World Wide Web (WWW), protein interaction networks, human metabolic networks, ecological networks and railroad networks are all complex networks [24, 116]. Scientists across disciplines including sociology, biology, computer science, linguistics and mathematics have identified the existence of many such networks in their domain. One common attribute to this assortment of complex networks is their massive size, mainly due to the large number of interacting individuals. A decade ago, analyzing the nature of these networks was a tedious task with survey tools and lot of manual interpretation. Recent advancements in computational techniques and data gathering has aided in the discovery, modeling and analysis of these networks without human intervention [87].

1.1 New science of networks

The study of networks dates back to 1736, when Leonhard Euler published the *Königsberg bridge problem* and its solution. Such networks were primarily referred as graphs and the branch of discrete mathematics that deals with these networks is *graph theory* [26, 46, 48]. The topological and structural properties such as planarity and isomorphism, representation of data in graphs, and the several types of graphs and trees along with their characteristics are

dealt with comprehensively in graph theory [46]. But, Complex networks form the class of graphs that exhibit non-trivial topological structures. The presence of edges in these networks is neither purely regular nor purely random. Viewed as large, dynamic graphs, their properties differ significantly from that of the classical *Erdös-Rényi graphs* [58]. There is no single accepted definition for complexity in networks [17, 32]. The dynamicity and the massive size of these graphs deter the application of theorems and lemma from classical graph theory. In general, the networks with thousands or millions of nodes and whose structure is irregular, complex and dynamically evolving in time are classified as complex networks [24, 41]. For example, the protein interaction of Yeast [84] is shown as a network (biological complex network) in Figure 1. Proteins are represented as nodes and their interaction is given by the edges. The network has 1870 nodes and 2240 edges.

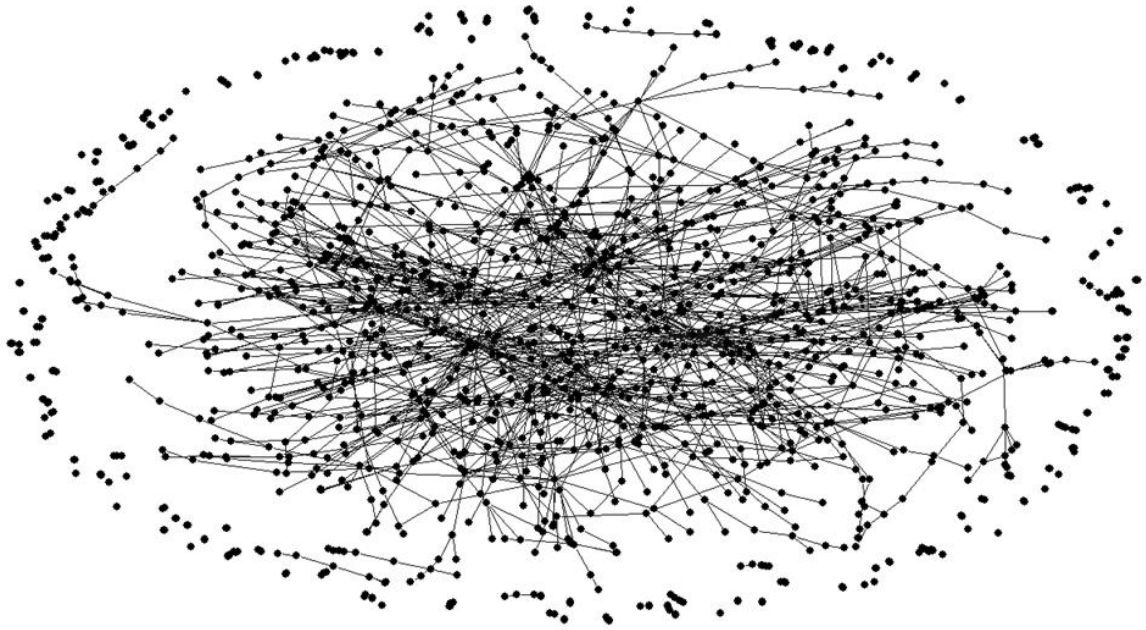

Figure 1: Protein interaction network of Yeast.

The study of complex networks further goes on to help unravel their correlation to the actual random graphs, and exploration of the macroscopic (global) and microscopic (node level) properties. Despite the multitude of research on complex networks, concentrating mainly on their topological and statistical properties, the knowledge about these networks is still considered to be in its infancy. The dramatic growth of the World Wide Web (WWW) and the statistical similarities of the web-graph with networks in other disciplines are the primary reasons for the sudden surge of interest in the study of complex networks. Hence these networks are sometimes referred as *web-like* networks [34]. The emergence of such networks has marked the beginning of the "new" science of networks [14, 164]. Researchers across disciplines have posted and answered several questions related to understanding the complex networks. The diverse nature of these networks, combined with their unique properties, is the main reason for the plethora of network related publications in the last decade. The U.S. military has appointed a separate cross-disciplinary research group to study the significance and impact of complex networks on the society [3]. The value of understanding these networks both on the academic front and in the social context cannot be overstated.

1.2 Community in graphs

One of the most significant properties that the nodes of any complex network exhibits is the tendency to be associative among group of similar nodes. Subsets of nodes, within a complex network, that have high affinity among themselves leading to *group-wise dense* connections. They also tend to be dissociative towards the remaining set of nodes. In other words, these subsets of nodes reveal higher interaction as compared to the entire system. Such *induced*

subgraphs have relatively high concentration of edges among themselves indicating their significance in the overall network. The existence of dense subgraphs in large sparse graphs, provide interesting insight to how these networks evolve because of the birth and death of nodes. The addition of a new node in a complex network does not randomly choose an existing node to connect. Instead, the node that is more influential in the network has a tendency to pull the new node towards itself (*preferential attachment* [51]). For example, a new webpage created will have hyperlinks pointing to the most popular and relevant websites on the same subject [37]. Similarly a new technical publication will definitely include the most influential articles, in its citation list.

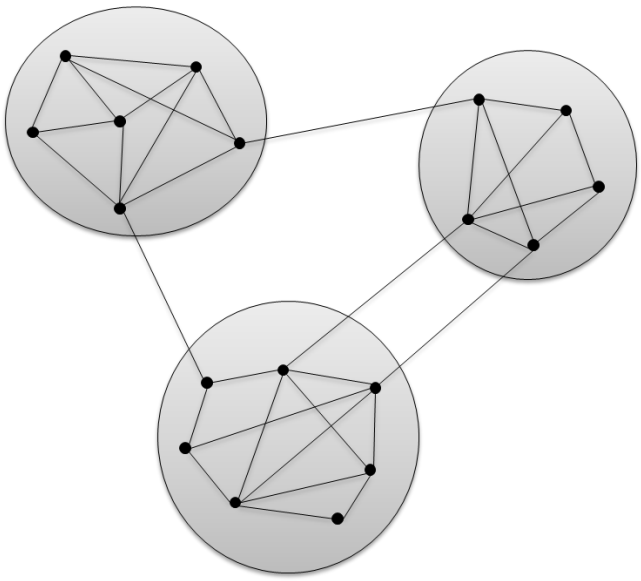

Figure 2: Communities in a graph.

In general, this phenomenon accounts for a closer bond amongst a group of nodes. The induced subgraphs thus obtained are termed as *Communities* (also called as *Clusters* or *Community structures*). Motifs, modules, clusters and hubs are other terminologies in the

literature referring to the subset of nodes closely-knit. As noted earlier, the nodes that form the community are tightly coupled with nodes that belong to the community (associative) and have fewer connections to the outside network (dissociative). Figure 2 shows an example graph with three communities (shaded regions).

Problems related to community discovery in networks are multi-fold. Some of the actively-pursued problems pertaining to communities are: detecting the number of communities in a given network; identifying the strength of the community relative to the global structure; identifying the membership of a specific node to a community; the importance of a given node in the overall network and its community (node centrality). Numerous algorithms exist in the literature to answer these questions in full or in partiality and some problems may still remain unexplored.

CHAPTER 2: COMPLEX NETWORKS

2.1 History of complex networks

Sociologists initiated the concept of connectivity and interaction among individuals in a society, when they studied the patterns of friendship and acquaintances in a neighborhood. Their results and observation were based on labor-intensive surveys. The outcome of these surveys had to be transformed into a concise representation in order to recognize the structure of the social neighborhood. Karinthy, F. [85], in his short story named "Chains", was the first to propose the concept of *five degrees of separation*. He supported his claim by showing five intermediate individuals (well-known to their immediate neighbors) were sufficient to connect a Nobel Prize winner and himself. Milgram performed a similar experiment to find out the number of acquaintances between two random people in the world [108]. Many such sociological experiments provided evidence that the links in a social network, though random, follows a non-random mold to its organization. This stimulated interests in the science of social connections [102, 144, 166].

2.2 Random graphs

Emergence of networks of genes, ecological systems and the spread of epidemic diseases extended the concept of random association from sociology to biology. Solomonoff and Rapoport were the first to systematically generate a synthetic network which replicated biological and social networks [148]. In particular, they were interested in knowing the

connectivity patterns (strong or weak) in random nets of neurons and epidemics. Their work marked the beginning of generating *random graphs*.

However, it was Erdös-Rényi's publication on random graphs [58] that laid foundation for the science of modeling synthetic networks and analyzing their properties. Given a fixed number of nodes *n* and edges *m* ($m \leq n(n-1)/2$), the Erdös-Rényi random graph (ER graph) construction starts with the null graph and adds an edge between two random nodes (if there is no edge already) one at a time. The step-by-step generation of an ER graph with $n = 5$ and $m = 5$ is shown as an example in (a) through (f) of Figure 3.

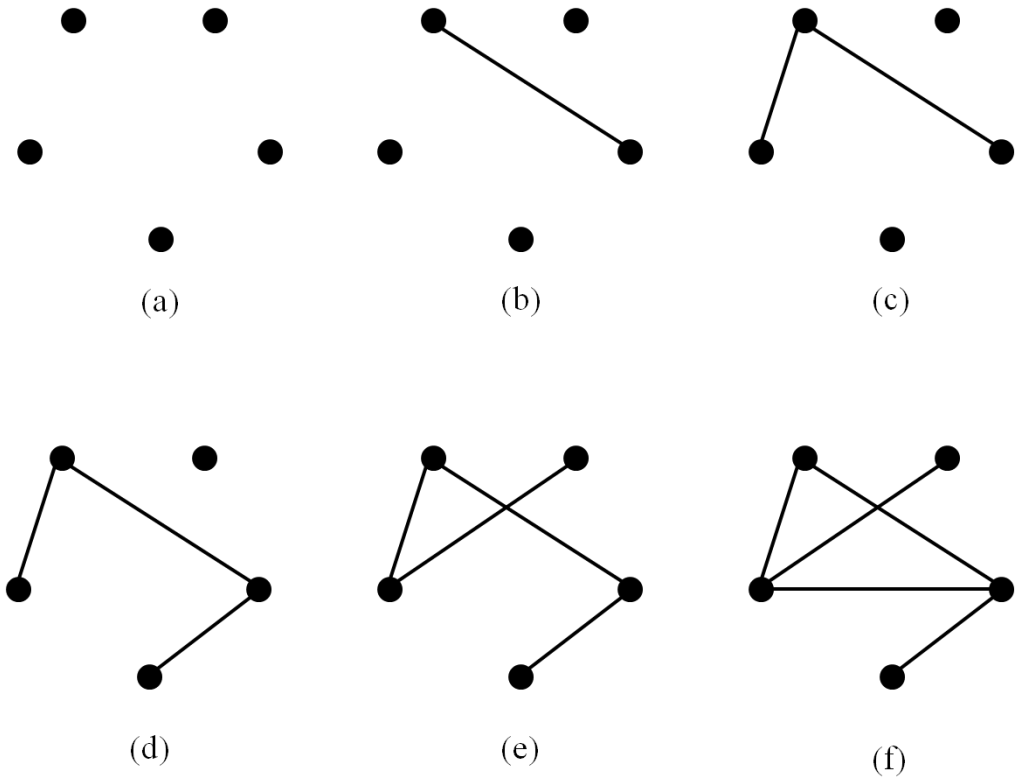

Figure 3: Erdös-Rényi random graph generation with $n = 5$ and $m = 5$.

Another popular random graph model proposed by Gilbert also has a similar technique, but it involves creating graphs with given n and p, where p is the probability that two given nodes have an edge between them [71]. So with $n = 5$ and $p = 0.5$ one can obtain the same random graph in Figure 3 with Gilbert's technique.

It is quite natural to include complex networks in the class of classical random graphs because they evolve in a random manner. But structurally, complex networks exhibit properties notably different from a random graph and thus require a separate branch of science investigating their characteristics. The differences include the dynamicity of the graph (birth and death of nodes), the associative nature of the nodes and non Erdös-Rényi randomness in the addition and removal of edges. Compiling the above properties together, a complex network maybe defined as a large, dynamic, random graph with non-uniform degree distribution and high clustering.

The growth of the internet and the WWW, and the advent of online social networks such as Facebook, Myspace, Linkedin, etc., has triggered interest in the study of complex networks. The statistical similarities these networks share with other social, biological, and linguistic real-world networks has increased stakes in models to simulate these networks, and techniques to understand their network properties.

2.3 Taxonomy of complex networks

Complex networks are ubiquitous and are of great interest in various disciplines - in graph theory, physics, statistics, sociology, biology and linguistics [49]. Due to their existence across several domains they tend to inherit certain properties inherent to the system they are modeled from. But the most interesting ones are their common characteristics and that clearly

demarcates their presence in a separate class of network science. In order to generalize the characteristics and properties of these networks, it is essential to categorize them and explore their features individually. The basis of classification of complex networks can be twofold: (*i*) based on their properties [10, 75] or (*ii*) based on their origin [49]. The properties that are common to most complex networks are of primary magnitude (as will be seen in the next section) and have been studied in greater detail by sociologists, biologists and mathematicians [50, 88, 118, 143]. A novel classification based on the origin of complex networks is presented and discussed in the following section.

2.3.1 *Natural vs. Manmade networks*

Complex networks can be broadly classified into *natural* or *manmade* networks based on their origin. As the name suggests natural networks are a result of modeling real-world complex systems that have entities interacting naturally. Most of the biological and social networks naturally occur and evolve in the real-world. So, they form the major sub-categories of the class of natural complex networks. Residential factions, social clubs and cultural groups are social networks formed naturally based on interaction among individuals. Any network that has human beings as nodes and depicts their interaction (sexual contact [22, 100], acting together in a movie [6], etc.) form a natural social network. Biological networks, which typically represent interaction among proteins, genes or metabolic molecules, also belong to the natural complex network category [39, 84, 124, 150]. Thus, within these two primary classifications, the natural networks fan out based on the individuals belonging to the networks. Ecological systems such as

the predator-prey network [153] and the network depicting the spread of epidemic diseases [57, 91, 99] are also types of natural biological networks.

On the other hand the information networks (e.g., the WWW, the Internet, e-mail network) [28, 60, 152] and the transportation networks (e.g., network of airport connections, road network) [138] are the result of the engineering technology. Apart from these human-engineered networks, the manmade networks can also include the computer-generated graphs to replicate the features of complex networks. These graphs are referred to as *synthetic* complex network models.

The man-made networks require more investigation and analysis because of their wide variety of applications. The three major types of artificial complex networks are the ones resulting from *engineering*, *literature* and *social networks* (*online*). Citation graphs [23] and word association networks [121] are the primary examples of linguistic (literature) complex networks. The significance of a published article with respect to its citations and peer publications can be inferred from a citation network [134].

The engineering class of complex networks can further be divided into *information*, *transportation* and *business* networks. *Information* networks are a result of modeling the communication systems such as phone calls, emails and blog/tweet follows as networks. The Belgian mobile phone network [23], University email network [155], and UK and Stanford web graphs [52] are all examples of information networks. Analyzing such networks reveal vital hints on the communication, social behavior and economic patterns on the individuals connected by them. Networks representing the flights operated between different cities, rail routes across the country and road transport across towns form the *transportation* networks [138]. Traffic

density and economic transits are some of the valuable information inferred from these networks. Network of power grids [53] and co-purchase [44] form the class of *business* (or financial) networks. One of the major beneficiaries of studying the properties of the business networks is *ecommerce* as they can enhance user preferences by better understanding their purchasing behavior through these networks.

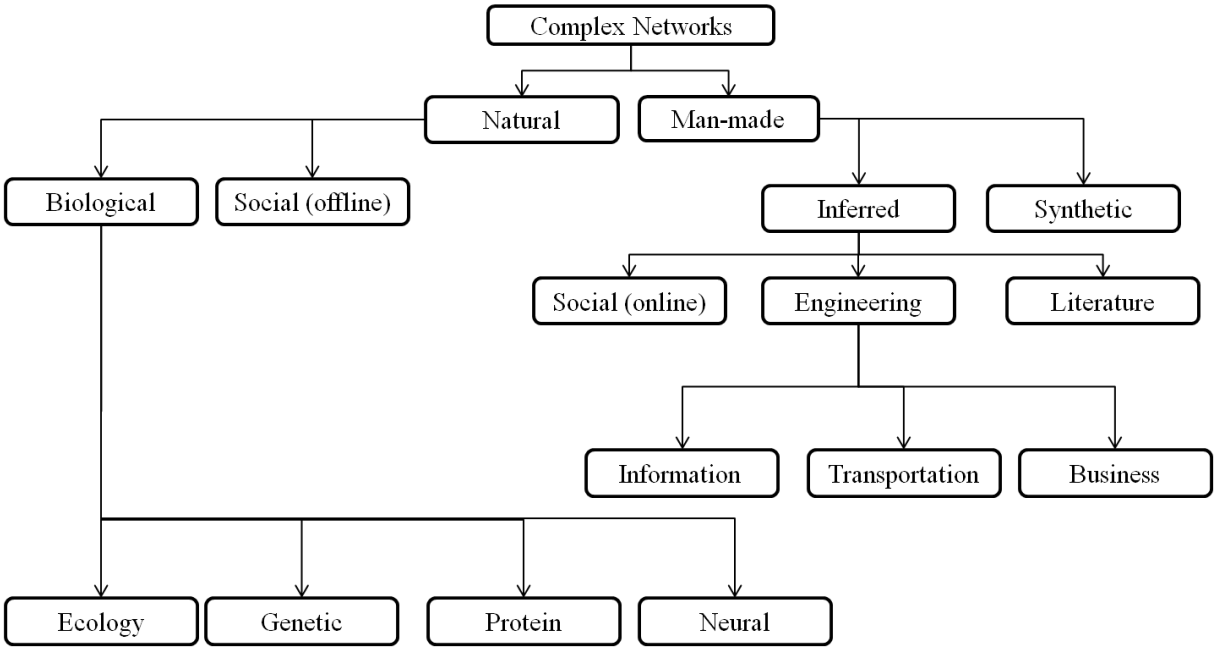

Figure 4: Taxonomy of complex networks based on their origin

2.3.2 *Online social networks*

One class of complex network that cannot be strictly classified under any of the above mentioned categories is the Social network [87]. Traditional methods to learn about the social links in a residential locality, or a school, or a tribe in Africa (for example) required survey tools and manual intervention to interpret the obtained data. These networks appear, evolve and may even vanish over a period of time, due to several factors such as migration, natural disasters,

industrial growth, etc. But their dynamic behavior takes place without any central authority governing them. So it is intuitive to classify them as a natural complex network. Recently, articles have focused on networks such as the Online Social Networks (OSNs) [98] and blog networks [77]. These networks are a part of the information networks and are easily accessible because of the transparent nature of the Web. Studying their structural properties reveals their interests and interaction based on addition and removal of nodes. They possess characteristics very similar to the naturally formed social networks. The only significant difference is that their interaction occurs through the Web and the Internet. Therefore we tag them under the class of *online social networks*.

2.4 Properties of complex networks

Despite randomness in their initial formation and unsystematic evolution, the complex networks result in a predictable aggregate. Be it the distribution of neighbors of each and every node or the grouping of similar entities, there is an emergence of order from chaos [3]. In recent times, several authors have provided brief descriptions on the traits of a complex system and differentiate them from other complicated systems that carry similar intricacies [15, 17, 55, 168]. Complex networks are indeed characterized by their unique attributes [9]. The structural and topological properties of complex networks are discussed mainly at two extreme levels in the literature: (*i*) *microscopic* - properties at node level (e.g. degree distribution, clustering coefficient), and (*ii*) *macroscopic* - global properties such as the average network distance (small-world effect). However, recent research has also focused on the *mesoscopic* properties which lie intermediate between these two, which is discussed in the following chapters.

2.4.1 *Small-world property*

One of the interesting properties to observe in a given large, random graph is the network distance between two random nodes. The network distance, also referred as *degree of separation*, is the minimum number of intermediate nodes (shortest path) required to connect two nodes. The average network distance is of particular interest in case of a random graph, because it determines the overall graph density. The *diameter* of the graph (longest shortest path) has also been investigated in the literature [25]. Milgram's experiment to find out the degrees of separation [108] marked the beginning of analyzing connectivity and interaction among individuals in a social network. He had requested random individuals (about 200) from Wichita, Kansas and Nebraska to send out similar packages, all with the same destination address - a doctor in Boston. The package can be sent(or passed on) from person A to person B if A knows B on a first name basis. Surprisingly, most of the packages reached the destination and the average number of intermediate hops between the origin and the destination was about six. Therefore, he concluded that two nodes are connected by an average of six intermediate nodes, now referred to as the *six degrees of separation* [164].

The Web graph also revealed similar phenomenon [7, 29]. The average distance between any two nodes of the web graph was observed to increase only logarithmically to the size of the graph. Hence, the large and sparse nature of any network does not imply a weak connectivity. This effect is termed as the *small-world* property [19] and has been observed in almost all complex networks [8, 19, 31, 56].

2.4.2 *Power law degree distribution*

Exploring macroscopic properties in networks with millions of nodes is tricky. Node level properties (microscopic) are much easier to corroborate empirically and they determine the robustness of a network. The number of edges incident on a vertex is called the *degree* of the node. The removal of a node with relatively higher degree would distort the topological nature of the network, whereas the death of an isolated node impacts the structure of the node less significantly. For example, in case of a social network, one or two individuals with high connectivity (degree) within a particular group influence the society to a greater extent. Another measure, that is closely associated with degree of a node is its *centrality* [68, 134]. The tendency of a node to be at the center or the quality of it being central denotes its *centrality*. Centrality determines the importance of a node in the network.

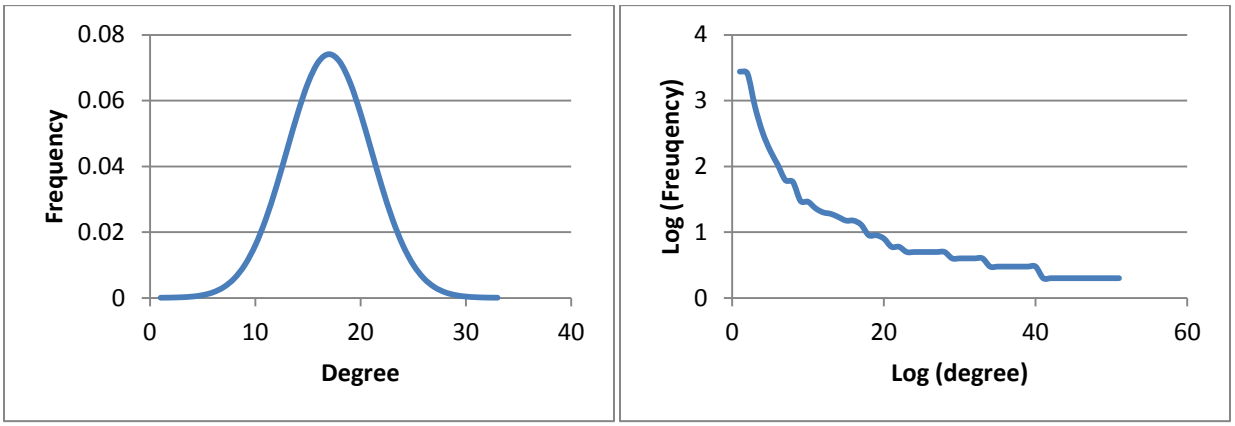

Figure 5: Comparison between *normal* (left) and *power law* (right) distribution.

Though, these measures are vital in determining the presence of a node relative to its neighbors, one metric that gauges the statistical characteristics of any network is the *degree*

distribution. In case of a classical random graph, every edge has equal probability of occurrence, thus the degree of the nodes follow a *uniform distribution* (Figure 5). The plot shows the uniform distribution shown by a random graph generated using ER random graph generator with 0.5 probability and 50 nodes. Whereas, in a complex network the nodes attach preferentially to few influential nodes and this leads to a rich-get-richer phenomenon. In other words, the nodes exhibit a *power law degree distribution* (Figure 5). The Power law curve is shown for an instance of the Web graph (10000 nodes) given by Albert *et al.* [7]

Given an undirected graph G (V, E) and a non-negative integer r,

$$P(r) = \frac{|\{i \in V \mid d_i = r\}|}{|V|} \qquad (1)$$

i.e., P(r) is the proportion of nodes of degree r in G. The degree distribution of G follows a power-law, if P(r) α $r^{-\gamma}$, where γ is the power-law coefficient [7, 88, 114]. This is also referred as *Pareto distribution*, because 20% of the nodes are highly connected to 80% of the remaining nodes.

2.4.3 *Clustering coefficient*

Power law degree distribution and small-world phenomenon are unique statistical properties of complex networks, but clustering of *associative* nodes and lower interaction (or dissociativity) among other nodes portrays the structural uniqueness in these networks. Subsets of nodes in a sparse complex network tend to connect more within their group than with nodes outside the set. In a social network, two people who have a friend in common are likely to become friends themselves. Such triangles are quantified by a metric known as *clustering*

coefficient [34, 115, 160]. This measure was first formally described as a parameter by Watts and Strogatz [165]. The clustering coefficient value of a node *u* is given by the equation

$$CC(u) = \frac{no.of\ edges\ between\ the\ neighbors\ of\ u}{d_u(d_u - 1)/2} \qquad (2)$$

In case of nodes with degree less than two, the value is zero. The clustering coefficient of the overall graph is the average value of the above equation for each node in the network and is given by

$$CC(G) = \frac{\sum_{v \in V} CC(v)}{n} \qquad (3)$$

Clustering coefficient is a normalized metric with its values lying between zero and one, with higher values denoting denser clusters. Almost all complex networks have a higher value of clustering coefficient (0.25 - 0.75). This is much higher when compared to that of the classical random graphs. A similar metric called *transitivity* dealing with the fraction of triangles, especially in small-world networks, was proposed by Barrat and Weight [18]. Transitivity of the graph G is given by

$$T(G) = \frac{3\ X\ number\ of\ triangles}{number\ of\ paths\ of\ length\ two} \qquad (4)$$

Table 1 summarizes these three properties and their corresponding values in case of a complex network and as a comparison the random network values are provided. The value of these parameters for a movie-actor collaboration network [6] is shown in Table 2.

Table 1: Complex network properties

	Average Distance	**Clustering Coefficient/Transitivity**	**Degree Distribution**
Complex Networks	O(log *n*)	high (>0.25)	Power law $P(r) \alpha r^{-\gamma}, 2 < \gamma < 3$
Random Network	O(*n*)	low	Uniform distribution

Table 2: Movie-actor network properties

Example	***n***	***m***	**Average Distance**	**Clustering Coefficient**	**Degree Distribution**
Movie-actors	449,913	25,516,482	3.48	0.78	2.3

The outcome of observing these properties and their corresponding values on a complex network is the evidence on the presence of cohesive subgroups. Within these sparse networks exist subset of nodes with higher clustering coefficient and their network distance is relatively smaller compared to the overall network. Each of these closely-knit groups includes one or more nodes that lie in the top 20% of the degree sequence, i.e. the nodes with higher degree. These nodes are generally referred to as *hubs* [4, 61] or *influential nodes* [76, 123] and they attract the nodes with not so high degree to form such dense clusters. The following section provides an elaborate discussion on these closely-knit subgraphs in a network, their complexity analysis [69] and a classification on the different techniques in the literature to explore them.

CHAPTER 3: COMMUNITY DETECTION

Closely-knit subgraphs, naturally occurring in complex networks, have been studied and explored to a great extent recently. Such dense subgraphs in large sparse graphs are termed as *Communities*. *Community detection* is the branch of complex network science dealing with the characteristics, definition, extraction and identification of such close-knit nodes. Network properties such as the average distance, degree distribution, clustering coefficient and transitivity are easier to compute despite the large nature of these networks. As mentioned in the previous section, these values are either node-specific or globally computable. The empirical values of these properties have been listed for over 700 real-world networks [120]. However, detecting subgraphs with certain properties in any given graph is relatively difficult because of their intermediate presence. The recent surge of real-world complex networks especially the social and biological networks has generated high interests in the algorithms to detect, extract and identify communities.

3.1 History of community

Community structure analysis began very early among sociologists who were interested in factions in local societies. They manually surveyed individuals of a society in order to find groups of people with similar interests. For example, Rice investigated the blocs in political bodies [136] and elucidated on the indices that can be used to identify and measure the strength of a political party in a given legislature. Weiss and Jacobson used sociometric techniques to study complex social systems and they analyzed the overall structure of a complex organization [166]. Later, the division among a group of students in a karate club was studied in detail by

Zachary [171]. The university-based karate club formally separated into two groups because of a dispute between the master and the instructor. The graph depicting this division among the students from Zachary's article is popularly known as the *Zachary Karate Club* graph. It is the most cited benchmark problem for community detection algorithms [11-13, 112, 113].

Biological networks are also prone to form such communities because of the inherent associative nature of the molecules and cells that they are comprised off. Clusters from molecular and protein interaction were shown to exist empirically [124]. Discussion on compartments in a predator-prey network also emerged [125]. The predator-prey structures, more commonly known as *Food Webs*, are not as large as the other biological networks [89] but still reveal dense subgraphs.

3.2 Related problems

Some authors refer communities in the same line as *clusters*. However, the term clustering emerges from *data mining* and it refers to extracting set of similar nodes from large datasets [79, 83]. Clustering concentrates mainly on the resemblance among the nodes using existing information, but communities investigate the strength of the connections between them. This differentiation set off a new literature dealing with community discovery algorithms., Procedures to uncover the closely-knit subgroups in social [67, 72], biological [72], business [133] and technological networks [62] evolved. Online social networks such as Facebook, Twitter, LinkedIn, etc. showcase this clustering property very naturally because of the active interaction among acquaintances, friends and friends of friends.

Mathematicians have been dealing with two similar problems in graph theory pertaining to extracting subset of nodes with significantly higher connections among them.

(i) *Graph partitioning*, refers to dividing the nodes of a graph into subsets of equal size, such that the number of edges or the weight of the edges (in case of a weighted graph) that fall between them are minimized [16, 86].

(ii) A *powerful alliance* is a subset of vertices such that each of them have more neighbors within the set to defend and defeat any attack from the neighbors outside the set [92]. Alliances in graphs can be of two types: (*i*) Given a graph *G* (*V*, *E*), a non-empty set $S \in V$ is a *defensive alliance*, if for every vertex *v* in *S*, *v* has at most one more neighbor in $V - S$ than it has in *S*. (*ii*) S is an *offensive alliance* if for every $v \in V - S$, that has a neighbor in *S*, *v* has more neighbors in S than in $V - S$. A powerful alliance is both defensive and offensive [27].

Community discovery is closely associated with these two existing problems in graph theory literature. They all focus on the property of the induced subgraphs. However there exist prominent differences between them. Extracting communities differs significantly from partitioning graphs because the latter divides the graph into equally sized partitions. Similarly, alliances concentrate on the relatively high degree of the nodes within the subset against their degree outside. But communities require more than just degree comparison. Finding optimal alliance in a graph is an NP-Complete problem [33]. Graph partitioning and community discovery problems are also NP-Complete [69, 142] since there is no polynomial time solution to verify the strength of each and every subgraph for a given graph.

3.3 Defining a community

Informally, a community maybe defined as a locally-dense subgraph in a large globally-sparse graph. An accepted mathematical definition for a community in graphs is yet to hit the literature, because it is difficult to exemplify its relatively dense nature [64]. For example, K_5, complete graph of five nodes, is a dense graph and will certainly qualify as a community if it forms a subgraph of a sparse graph. But if it is part of a well-connected component consisting of thousands of nodes, it is not significantly large enough to be a dominant structure. Existing definitions of a community are conceptual and depends mainly on the topology of the underlying network. The relation every node shares with its neighbors, both within and outside the subgraph forms the essence in those definitions. Some of the definitions are also constructive, i.e., the result of algorithmic steps [140]. We classify the community definitions in the literature into three categories and discuss them briefly in this section.

3.3.1 *Diameter*

Earlier notion of a community focused on its equivalence to a *Clique* [102] . But expecting every member to be connected to every other member within a subgraph of a sparse random network is stringent. Therefore, a more relaxed definition based on the maximum distance between two nodes of the subset was proposed. Instead of requiring every neighbor to be at a distance one [clique], the subgraph *diameter* [46] would have an upper limit of *d*. The *diameter* of a graph is defined as the largest distance between two nodes in the graph. The terms *d*-clique [5], *d*-clan and *d*-club [109] are some definitions resulting from these relaxations in the diameter and were used to describe communities. But a subset of nodes with a maximum

diameter *d* in a graph need not be connected, since there maybe shortest paths through nodes that are not in the subgraph. Thus, diameter based definition of a community fails to extract the densely connected subset of node.

3.3.2 *Degree*

Distance between nodes of a subgraph cannot thoroughly determine its relative denseness. But defining a community based on the *degree* of a subset of nodes is a reasonable approach. Nodes with high degrees tend to attract other higher degree nodes, thus leading to denser structures [88]. If the internal degree of every node in the subgraph is greater than or equal to δ (minimum degree of the graph), then there exists a stronger tie. In other words, every vertex is adjacent to at least δ other vertices within the subgraph. The terms δ-plex [145] and δ-core [146] refer to subgraphs satisfying such degree-based criteria and were referred to as communities. A more formal approach to defining communities was attempted by Radicchi *et al*. [129]. They defined a community as a subset of nodes whose internal degrees add up to a value greater than the sum of their external degrees. Later, Hu *et al*. [82] suggested a finer version of this definition, in which the external degree takes into account only those edges incident on the nodes in other communities. Both the definitions considered the resulting subset of vertices as a *weak-community*, because the number of internal edges will always be counted twice (because the sum of the degrees of a graph is twice the number of its edges), but the external edges are counted only once.

3.3.3 *Alliance*

In order to define a community in the strong sense, comparing the degree of the entire subgraph against its neighborhood is not sufficient. Instead, a stronger association is required among nodes within the induced subgraph compared to outside. Flake *et al*. [61] proposed that the internal degree of every vertex within a community should be greater than or equal to its external degree. Even though they suggested it in the context of a web graph, it was a widely accepted definition because such a set of vertices would form greater bond within the subset than outside. A narrower version of the definition was later suggested by Radicchi *et al*. [129] and Hu *et al*. [82]. They defined a strong community as the one in which each node has more adjacent nodes within the community than with the rest of the graph.

These definitions coincide with that of *alliances* in graph theory as described briefly above. Table 3 gives a brief listing of the various definitions for a community discussed in this section. It is evident that a community needs to be defined in terms of the degree of the nodes. Especially, in the case of a large sparse graph, a dense subgraph should have nodes with higher *average degree* than those randomly present. Chapter 6 discusses in detail a novel community definition based on this notion. The following section discusses the most cited community related metric called *Modularity*.

Table 3: Existing definitions of a community

Name	Definition	Reference
Clique	Every vertex is adjacent to every other vertex	[Clique] [102]
Diameter	diameter of $g \leq d$	[*d*-clique] [5] [*d*-clan & *d*-club] [109].
Degree	Every vertex is adjacent to at least δ other vertices in the subgraph [internal degree of every vertex $\geq \delta$]	[δ-core] [145]; [δ-plex] [146].
	Sum of internal degrees > Sum of external degree	[Weak-community] [129].
	Sum of internal degrees > the number of edges the subgraph shares with other communities	[Weak-community] [82].
Alliance	Internal degree of each vertex > External degree of the vertex	[Strong alliance] [129].
	Internal degree of every vertex is greater than the number of edges the vertex shares with other communities	[Strong alliance] [82].
	Internal degree of each vertex ≥ External degree of the vertex	[Defensive alliance] [61].
	Set of nodes such that each of its proper subsets has more edges to its complement within the set than outside	[LS-set] [101].

3.4 Modularity

Girvan and Newman were the first to formally define a community in social and biological networks. They described a community to be a subset of nodes with more connections between them and relatively fewer edges to the outside network [72, 117]. Flake *et al*. defined a community on similar terms, but in the case of a web graph [62]. A *web community* is a

collection of web pages such that each page has more hyperlinks to pages within the community than outside. These abstract definitions formed the base for discovering community in complex networks [156]. However, one of the most widely accepted mathematical definitions of a community in social network literature is the network modularity, proposed by Newman [113, 117]. *Modularity* (denoted by *Q*) measures the quality of a particular induced subgraph in a given network. Modularity for a network partitioned into *k* communities is given by

$$Q = \sum_{i=1}^{k} (e_{ii} - a_i^2), \qquad (5)$$

where, e_{ij} belongs to the *k* X *k* symmetric matrix and gives the fraction of the edges that link the vertices in community *i* to vertices in community *j*. The trace of this matrix $\sum_i e_{ii}$, is the fraction of edges that connect vertices within the same community. The row sum $a_i = \sum_j e_{ij}$, is the fraction of edges incident on all vertices in community *i*.

Modularity value has a range from zero to one. Values of *Q* approaching one indicate a strong community structure. If the number of within-community edges is no better than random, or if the given graph is considered as one community, the value of *Q* will equal 0. A complete graph K_n is the strongest community structure. But $Q(K_n) = 0$ according to the above equation. Whereas two complete components connected by an edge or two will have $Q \approx 1$. So, given a set of partitions, the highest value of modularity indicates a good community structure. In spite of the accuracy and popularity in identifying dense communities, modularity optimization techniques are not suited for large networks [11, 66].

CHAPTER 4: ALGORITHMS TO DETECT COMMUNITIES

Algorithms dealing with community detection in complex networks address one of the following two questions: (i) given a network, can we explore and extract subsets of nodes that form a community? (ii) given a network and a seed node, can we identify the best community that the given seed belongs to, if there exists one? The former problem, known as *Community Discovery*, has been studied extensively in the literature and a number of approximate algorithms has been proposed [44, 62, 72, 110, 137]. The latter, known as *Community Identification*, has also been studied in the literature, but to a smaller extent. Techniques to discover communities in complex networks have been broadly discussed by a few authors [64, 127, 141, 157]. A comprehensive taxonomy classifying the community detection algorithms in the literature is presented in this section. But prior to that the existing hierarchical clustering based classification of the algorithms and some of the key techniques in the literature are also discussed. They focus on classifying the techniques based on either *divisive* or *agglomerative* hierarchical techniques, but a novel classification based on their underlying technique is depicted.

4.1 Hierarchical clustering techniques

Most community detection algorithms are based on hierarchical clustering techniques, which extract clusters based on similarity metrics. The algorithms are divided into two classes, based on addition or removal of edges to or from the network: (*i*) In *Agglomerative* algorithms, the similarities are calculated between vertex pairs by some method and edges are then added to an initially null graph starting with the vertex pairs with highest similarity [117]; (*ii*) *Divisive* algorithms start with the given network and the edges with maximum "betweenness" are

computed and eliminated at each step. Both procedures can be halted at some suitable points and the resulting set of connected components form the communities. The following subsections elucidate these two techniques and give a detailed view of a new taxonomy of community detection methodologies.

4.1.1 *Agglomerative*

As mentioned earlier, agglomerative algorithms buildup to a graph with communities from a null graph. The iterative addition of edges is based on the value of a similarity metric. Number of node-independent paths, edge-independent paths or paths that run between the vertices are some of the examples of metrics considered in a hierarchical agglomerative algorithm. These algorithms are good at discovering the strongly linked cores of communities [117]. As and when the edges are added the procedures start building a tree from an empty set of vertices (leaves), which can be represented by a *Dendrogram* [72] as shown in Figure 6.

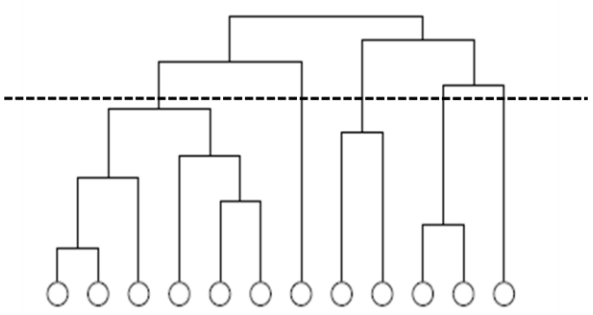

Figure 6: Dendrogram

Some of the popular agglomerative techniques in the literature are explained below

Graph-Partitioning methods

Graph partitioning requires dividing its vertices into a number of disjoint sets of roughly equal sizes, while minimizing the number of edges incident between the vertices in different sets [110]. Community discovery algorithms base themselves on traditional graph partitioning [86, 128].

Spectral bisection: The Laplacian of an undirected graph is a symmetric matrix L = D – A, where the diagonal element $l_{ii} = d_{ii}$ is the degree of the i^{th} vertex; and the off-diagonal element $l_{ij} = -a_{ij}$ is the negative of the corresponding element in the adjacency matrix A. If the network separates perfectly into communities (i.e., into *g* components) then the Laplacian will be a *block diagonal*. There will be *g* degenerate eigenvectors with eigenvalue 0 [128]. If the division of the network is not components, then there will be *g* eigenvalues slightly different from 0 (non-negative). This method is reasonably fast, but only bisects the graph and to obtain more partitions or larger number of communities the spectral bisection is applied repeatedly to the subdivisions.

*ComTector**:*** *Cliques* and near-cliques correspond to high connectivity within a given graph. *ComTector* builds communities around *overlapping cliques*. The *maximal cliques* are considered to be clustering kernels attracting other nodes towards them. They can result in denser communities. The larger the size of a clique, more likely a kernel it would become a community structure [53]. So, the cliques are arranged in descending order and the smaller ones are discarded. The cliques are chosen one at a time iteratively from largest to the smallest and removed at each step. Each maximal clique that contains the centers of the remaining elements in the set is removed from the clique. Thus, at each step the kernel enclosing other smaller kernels

are identified and removed. On repeated removal the dominating kernels are identified as the communities.

Kernighan-Lin algorithm: The Kernighan-Lin partitioning algorithm is a greedy optimization method that assigns a benefit function to a partition of the network and then attempts to maximize the value of that function [86]. This algorithm also *bisects* the network as in the spectral bisection method and requires the user to specify the number of nodes in the two subsets. It is a two stage algorithm in which the first stage involves finding the change in the benefit function when the one vertex from each subset is chosen and swapped. The second stage scans the sequence of swaps made to find the one with the highest value and chooses this to be the bisection of the graph. Specifying the size of the partitions as a priori (thus making it unsuitable for real-world networks) and repeated bisection of the graph are the principal drawbacks of the method.

Label propagation algorithm

Raghavan *et al*. [130] proposed a linear time algorithm that does not require any prior information about the number of communities. The algorithm is analogous to an *epidemic* in a locality where the individuals acquire disease most prevalent in his or her neighborhood [98]. The iterative algorithm initializes each node with a label and updates the label in each run with the maximum occurring label amongst its neighbors. If the label represents a community which the node, say *x* belongs to, then the neighbors with the same labels will influence *x* to belong to their community and hence the name *label propagation* algorithm (LPA). The iteration and

updating of labels continues till no node in the network changes its label. In other words, the iteration stops when every node has a label that the majority of its neighbors have.

There are two variants of the label propagation algorithm – *synchronous* and *asynchronous*. Synchronous label propagation updates the label as and when it finds a new label and makes it available for the remaining nodes during the iteration. The asynchronous version waits for all the nodes to decide on their labels in an iteration and updates before the next iteration. An improved version of the LPA was proposed by Leung *et al.* [98]. This algorithm provides stability to the communities by avoiding labels spreading to large number of nodes initially. The extension of the algorithm adds a score associated with each label whose value decreases as the label propagates. The performance of this algorithm betters the original as a result of balancing the cluster formation at each step of the algorithm.

CNM algorithm

Clauset, Newman and Moore (*CNM*) proposed a hierarchical agglomeration clustering approach [44] that performs a greedy optimization of the modularity metric *Q*. While finding the global maximum modularity over all possible divisions is hard, approximation techniques have been proposed which gives reasonably good solutions [13]. The algorithm uses two quantities for calculating modularity:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw}\, \delta(c_v, i)\delta(c_w, j) \qquad (6)$$

$$a_i = \frac{1}{2m} \sum_{v} d_v\, \delta(c_v, i), \qquad (7)$$

where A_{vw} is the adjacency matrix value corresponding to nodes v and w, δ function $\delta(i, j)$ is 1 if $i = j$ and 0 otherwise; $m = \frac{1}{2}\sum_{vw} A_{vw}$ is the number of edges in the graph, d_v is the degree of the node v which is assigned to community C_v. The algorithm requires finding the changes in Q resulting from the amalgamation of each pair of communities and choosing the largest.

Clauset, Newman and Moore also suggest improvements to the data structures that provide considerable saving to memory and time. The CNM algorithm uses two data structures to find a community pair with maximum ΔQ (change in modularity) value: (1) a balanced binary tree (or a heap) which stores the community pairs and (2) a max heap that is sorted by the ΔQ values. Wakita and Tsurumi [161] observe that the algorithm does not scale, and performs well only for mid-scale communities (fewer than 500 nodes). They propose an alternate metric known as *consolidation ratio* given by,

$$ratio(C_i, C_j) = \min\left(\frac{|C_i|}{|C_j|}, \frac{|C_j|}{|C_i|}\right), \quad (8)$$

where $|C_i|$ is the size of the community i in terms of the number of links to other communities. The ratio controls the growth of the community in a balanced way and considers merging the communities C_i and C_j based on the product of the ΔQ with the ratio value.

Bibliometric approach

Motivated by the metrics used to define the similarity between scientific publications, *bibliometric coupling* and *co-citation coupling*, Balakrishnan and Deo proposed an algorithm that computes the *bibliometric similarity* between all pairs of vertices in the given graph [13]. Using the metric computed on n isolated nodes, edges between pairs of nodes starting with the

pair of the highest similarity is introduced, progressing to the weakest. The measure of similarity between two nodes *u* and *v* is given by

$$\frac{|N[u] \cap N[v]|}{\min(d_u, d_v) + 1}, \qquad (9)$$

where $N[u]$ is the closed neighborhood of node *u*, and d_u is its degree. One result observed from this technique is that algorithms employing local properties of the graph seem to produce better community partitions than the ones employing the global properties.

Potts model

A physics inspired approach to community detection, referred to as *q-potts model* involves assigning a *spin state* between 1 and *q* at random to each node. *q*-potts model is an *Ising* model with *q* states instead of just 0 and 1 [135]. The algorithm consists of: (1) assigning spins randomly to a pre-selected number of communities, (2) selecting a spin and calculating the energy change as the spin is moved or added to another cluster and choosing the one which yields the maximum energy, (3) exchanging and calculating energy by moving spins till a local energy minimum is reached. The above steps are repeated by sampling initial random configurations and the best is chosen [139]. The energy of the spin system is given by the Hamiltonian

$$H(\{\sigma\}) = -\frac{1}{2}\sum_{i \neq j}\left(a_{ij}A_{ij} - b_{ij}J_{ij}\right)\left(2\delta\left(\sigma_i, \sigma_j\right) - 1\right), \qquad (10)$$

where A_{ij} is the adjacency matrix elements, $J_{ij} \equiv (1 - A_{ij})$ and a_{ij} and b_{ij} are the positive weights of the connected and unconnected edges respectively, σ_i is a Potts spin value $1 \leq \sigma_i \leq q$,

that designates the specific community, and function $\delta(i, j) = 1$ if i = j and 0 otherwise. This method predicts fuzzy communities which are clearly separated from others and since uses only local values to calculate Hamiltonian and update spins proves to be a faster algorithm.

4.1.2 *Divisive*

Divisive algorithms iteratively remove edges from the given graph and are also two step processes for extracting communities; (1) calculate the similarity metric and (2) remove the edge between the least similar pair of nodes. While agglomerative algorithm builds the dendrogram from the leaves upwards, divisive techniques start at the root (top) and disintegrate the graph into communities by the removal of edges. The fundamental ingredient of a divisive algorithm is a quantity that can single out edges connecting nodes belonging to different communities [129].

Betweenness score

Edges that connect two communities would be included in all *paths* between them. The divisive algorithm proposed by Newman and Girvan [72], eliminates edges with highest betweenness score (*GN algorithm*). The focus is not on removing the vertex pairs with lowest similarity, but on finding edges with the highest betweenness and removing them. Examples of betweenness metric include shortest path, random-walk and current-flow, and have been discussed extensively [117]. After the removal of an edge the betweenness score is recalculated for the new graph and the edge with the highest value removed. The algorithm thus iterates in calculating of the score and removal of edges.

The GN algorithm [72] was considered computationally inefficient since it involves calculating a global quantity, edge betweenness, whose value depends on the properties of the

whole network. Raddichi *et al*. [129] proposed *edge-clustering coefficient* as a metric to eliminate edges. It is analogous to the node-clustering coefficient, which denotes the fraction of pairs of neighbors of a node that are neighbors themselves. The edge-clustering coefficient is defined as the number of triangles a given edge (*i*, *j*) belongs to, divided by the number of triangles possible with that edge, given the degrees of the nodes *i* and *j*. In general, the edge-clustering coefficient of higher order cycles, say order *g*, is given by:

$$C_{i,j}^{g} = \frac{Z_{i,j}^{g} + 1}{S_{i,j}^{g}}, \qquad (11)$$

where $Z_{i,j}^{(g)}$ is the number of cyclic structure of order g the edge (*i*, *j*) belongs to and $S_{i,j}^{g}$ is the number of possible cyclic structures. An edge connecting two communities would have a lower coefficient value, due to lesser number of cycles. The 1 to the numerator avoids *zero triangles network*. This algorithm extracts better communities in larger networks since it is based on vertex properties.

Swarm aggregation

Nodes in a network behave analogously to particles in a self-driven system and this lead to the *swarm aggregation algorithm* proposed by Oliveira and Zhao [119]. The algorithm is adaptive, aggregates information from adjacent nodes and approximates node that belong to the same community. Each network node v_i has an initial angle value $\Theta_i(t)$ chosen in $[0, 2\pi)$. Then the angle value of each node is updated based on the angles of its neighbors using the following equation

$$\theta_i(t+1) = \theta_i(t) + \eta_i(t)\left[\sum_{j=1}^{d_i} w_{ij}\theta_j(t) / \sum_{j=1}^{d_i} w_{ij} - \theta_i(t)\right], \qquad (12)$$

where d_i is the number of neighbors of node i, $\eta_i(t)$ is i's moving rate at time step t and w_{ij} represents the influence of neighbor j in updating i's angle. The influence of the node's angle from its neighbor is based on two main factors; common neighbors (*CN*), proportion of shared neighbors and similarity (*SN*), similar neighbors, which is considered in the calculation of w_{ij}

$$w_{ij} = CN(i,j)\ X\ SN(i,j). \qquad (13)$$

If a community is composed of sub-communities, nodes belonging to the same sub-community quickly group themselves and fall apart from nodes belonging to another sub-community.

Information centrality

The algorithm, proposed by Fortunato *et al*., discovers community structures based on *information centrality* [66]. Procedurally, this is very similar to the betweenness score algorithm given by Newman [117]. Information centrality, C_i^I for an edge incident on node i is based on network efficiency F and is defined as the relative drop in the network efficiency caused by the removal of the edge. The efficiency $\in_{ij}$ between two nodes i and j is inversely proportional to the *shortest path length* d_{ij}. F is a measure of how efficiently the exchange of information in a network G (with n nodes, m edges) takes place through a node. The network efficiency of a graph G is the average of $\in_{ij}$ and is given by:

$$F[G] = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}. \quad (14)$$

The information centrality C_i^I of the node i is calculated as the relative drop in the network efficiency based on the removal of the edges incident on i:

$$C_i^I = \frac{\Delta F}{F} = \frac{F[G] - F[G']}{F[G]}, \quad (15)$$

where G' (n nodes and ($m - 1$) edges) is the network obtained from G with the removal of the edges incident on i. The method consists of finding and removing the edges with the highest centrality score until the network breaks up into components. These evolve from a communication network where information travels along the geodesic paths. The value is recalculated each time after the removal of an edge.

External optimization

This algorithm based on optimizing modularity value Q uses a heuristic search based on the external optimization (*EO*) algorithm [54]. The algorithm initially proposed by Boetcher and Percus [26] operates on improving the local variables (individual nodes) to optimize the global variable (Q). The equation of modularity can be written as

$$q_i = d_{r(i)} - d_i a_{r(i)}, \quad (16)$$

where, $d_{r(i)}$ gives the contribution of each node i (node degree) and a_r gives the fraction of edges that have one or both vertices inside the community, on placing the node into a certain partition

r. In order to find out the contribution of the node *i* to the modularity, relative to its own degree, the above equation is normalized in the interval [-1, 1] as given below

$$\lambda_i = \frac{q_i}{d_i} = \frac{d_{r(i)}}{d_i} - a_{r(i)}. \qquad (17)$$

The optimization works as follows:

1. Split the nodes of the whole graph in two random partitions (two communities) having the same number of nodes in each one.

2. At each time step, the system self-organizes by moving the node with the lower fitness (extremal) from one partition to the other. In principle, each movement implies the recalculation of the fitness of many nodes because the right hand side of equation involves the pseudo-global magnitude $a_r(i)$.

3. The process is repeated until an "optimal state" with a maximum value of Q is reached. After that, we delete all the links between both partitions and precede recursively with every resultant connected component.

4.2 Taxonomy of community detection algorithms

As mentioned before, community detection algorithms can be broadly categorized depending on whether they discover communities globally or identify a specific community locally. : (*i*) given a network, can we identify or extract sets of nodes that satisfy the community structure property?, or (*ii*) given a network and a small set of nodes, how do we identify the best community structure that includes these nodes, if there exists one?. The former problem, known as *Community Discovery*, has led to a number of algorithms. The latter, *Community*

Identification, has also been studied briefly in the literature. Figure 7 illustrates the taxonomy of algorithms classified based on these two branches of community detection.

It is evident from the taxonomy that the key techniques described above lead to many other algorithms. Some of the underlying techniques apart from the ones discussed above that lead to several algorithms to detect, explore and extract communities in networks are as follows:

- Betweeness approach [72, 77, 117, 129]
- Bibliometric technique [13, 81]
- Graph partitioning methods [16, 86].
- Label propagation techniques [45, 98, 130]
- Modularity based algorithms [38, 43, 44, 103, 111, 163]

Each of these methodologies has their pros and cons. Quite a few articles have addressed the shortcomings and limitations of one or more of these basic procedures [11, 65, 93]. Community discovery is explained in the following subsection. Since our research work focuses mainly on community identification, it is addressed in detail in the next chapter.

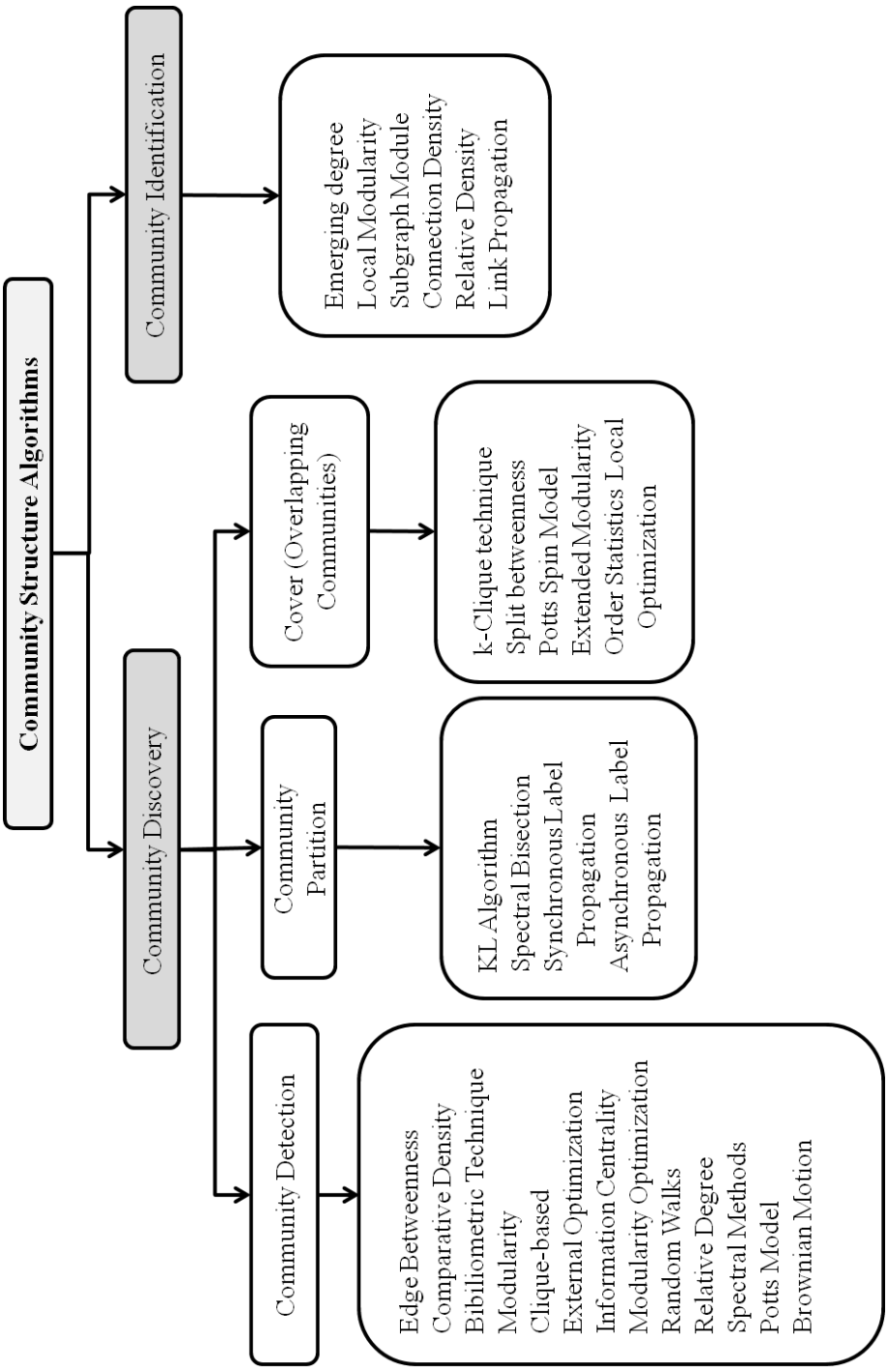

Figure 7: Taxonomy of community detection algorithms

4.2.1 Community discovery

Most of the algorithms related to communities in complex networks specialize in finding or extracting communities globally. Discovering communities in any well-known real-world networks or an artificial benchmark graph has been widely studied in the recent decade and several procedures have been proposed to extract such communities [64, 127, 141, 157]. The problem can be formally stated as follows. Given a graph G (V, E), classify the nodes V into k subsets, $C_i \subseteq V,\ 0 \leq i \leq k$; such that the nodes of each C_i have higher edge density relative to the graph. Community discovery can result in one or more communities in a given graph or the given graph need not feature any community at all. On similar terms, a node can belong to more than one community. Therefore, the classification of community discovery algorithms further branches down to the following: *Community Detection*, *Community Partition* and *Overlapping Communities* (*Cover*). These subclasses are formally defined as follows.

Detection

Is there a community present in the graph? If yes, C_1, $C_2 \ldots$ C_k strictly form the communities. Isolated nodes can exist and do not form communities. In other words every node can belong to at most one community. Community detection algorithms extract all the communities in a given graph and can leave out nodes that do not belong to any community.

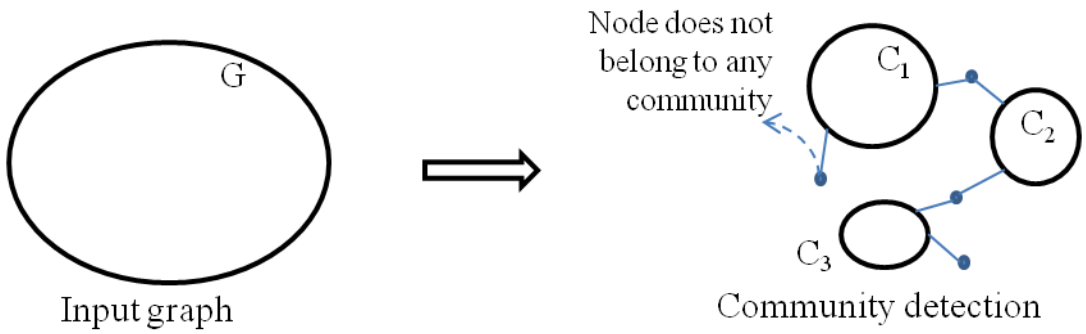

Figure 8: Community detection

Partition

C_1, C_2… C_k form communities such that,

$$C_i \cap C_j = \emptyset \text{ , if } i \neq j.$$

Any given node has to belong to a community (i.e. exactly one community).

This problem is different from the classical Graph-Partitioning Problem, since the number of communities and the size of the community are prior unknown [147]. Community partition algorithms extract all the communities in a given graph and all the nodes belong to one community or the other

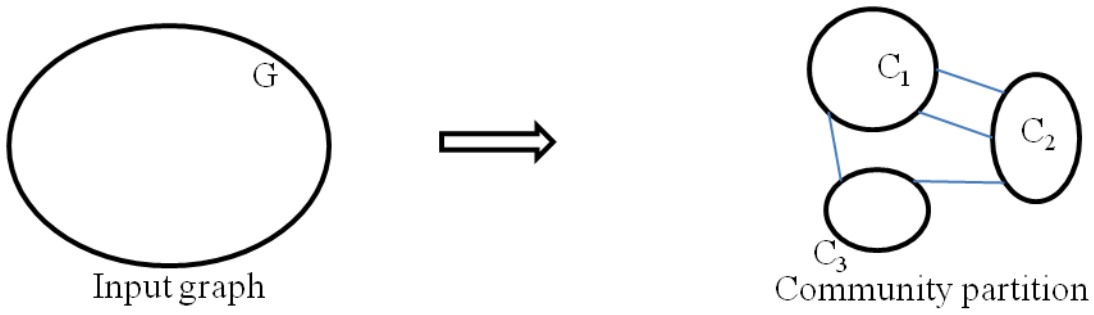

Figure 9: Community partition

Cover

A node can belong to more than one community but should belong to at least one community. If a node belongs to two or more communities, the algorithms refer to them as *overlapping communities* or a cover. Community discovery algorithms that extract overlapping communities result in each of nodes classified into one community at least.

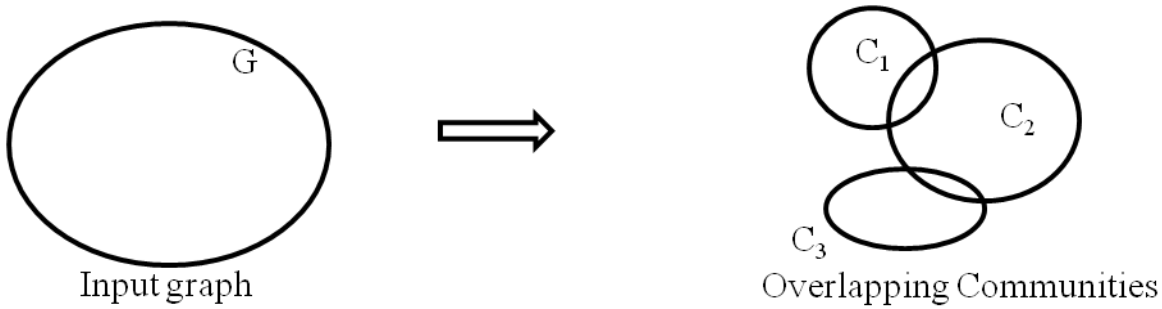

Figure 10: Overlapping communities.

4.2.2 Community identification

The exploration of a large complex network to discover subgraphs can be computationally expensive and challenging. But to identify one dense subgraph that a given node belongs to is relatively easier. Even though community identification requires exploring fewer nodes as against community discovery, it is still an NP-Complete problem. The subsequent sections discuss this problem in detail with examples and existing solutions in the literature, followed by novel algorithms for community identification. A proof by reduction to show the NP-completeness of the community identification problem is also given.

CHAPTER 5: COMMUNITY IDENTIFICATION ALGORITHMS

Community identification can be formally described as follows: Given a large sparse graph *G* (*V*, *E*) and a *seed vertex* $u \in V$ (or a set of vertices), does there exist a community that *u* belongs to? If yes, return the induced subgraph. Community discovery, on the other hand, deals with exploring all the communities in any given graph. But the problem that is of more interest in many real-world complex networks is the existence of community that any given individual node belongs to. Also, solving this problem is relatively easier since it does not require the knowledge of the entire graph. This leads to the class of algorithms for identifying a specific community to which a given node belongs. This section elucidates in detail the different type of nodes that constitutes a community and existing community identification algorithms. An improved metric to identify communities is also discussed.

5.1 Nodes of a community

In a large sparse graph, a node need not always belong to a community. Figure 11 shows a typical structure of a Community $C(V_C, E_C)$ in a given graph *G*. There are two subsets of the nodes that form a community, B_C and N_C, such that

$$B_C \subseteq V_C, N_C \subseteq V_C$$

$$B_C \cap N_C = \emptyset$$

$$B_C \cup N_C = V_C$$

B_C forms the *boundary nodes* of the community. These are the nodes that have neighbors both within and outside the subgraph *C*. The set N_C forms the *nuclei* of the community, which includes nodes that have neighbors only within the community.

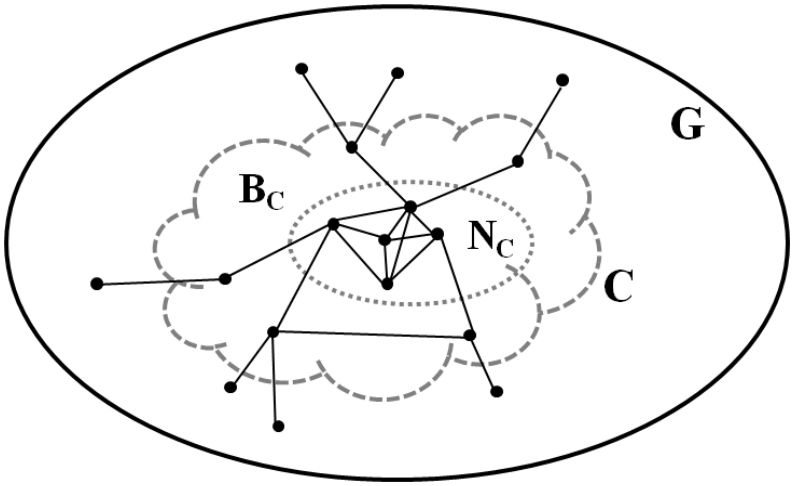

Figure 11: Components of a Community

The input seed node could be one of the nuclei or boundary nodes or a node that is more like a leaf. There are three possible scenarios based on the location of a seed vertex (Figure 12) and any community identification algorithm should address these scenarios. They are as follows:

(i) The seed vertex belongs to the *core* vertex set (nuclei) of a community. This is the primary scenario and the algorithm should easily identify the corresponding community, because the traversal covers all the nodes.

(ii) The seed vertex belongs to a community, but not to the core set. In other words, the seed vertex is one of the *boundary* vertices and thus can belong to more than one community. Such overlapping communities are dealt with in a separate class of algorithms [157]. But for community identification, when the seed vertex belongs to the overlapping zone, the algorithm should be able to identify at least one of the communities, possibly the densest one.

(iii) The seed vertex does not belong to any community and so the algorithm should not return any community associated with such a node.

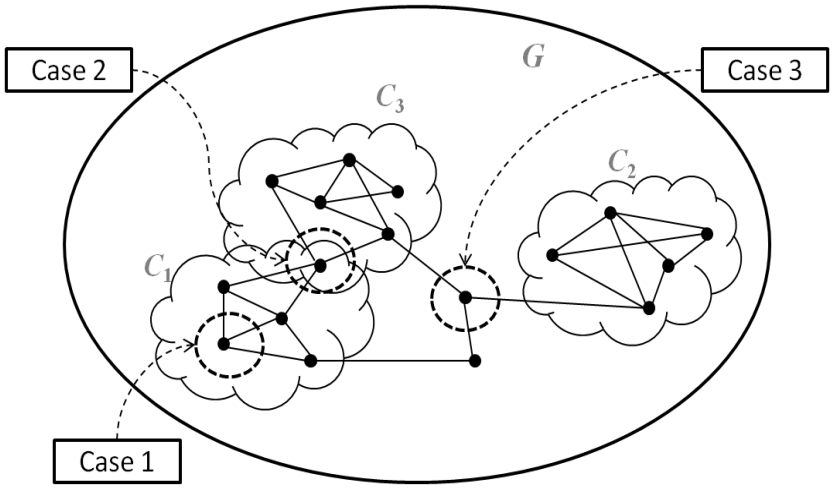

Figure 12: Community identification scenarios. C_i denotes a community.

5.2 Community identification preliminary

Given $G(V, E)$, any community identification algorithm would adhere to the following steps in general:

1) Begin with the seed vertex $u \in V$.
2) Locally traverse the graph from *u* only using its adjacency information. Breadth-First Search (BFS), Simulated annealing [142], etc., are some of the techniques to carry out the procedure.
3) Choose nodes to be included in the community.
4) Traverse further from the newly added nodes till no new nodes can be added.

Initially, the traversal begins with the seed vertex's adjacency information. Then the neighbors are locally explored to reach out to the next set of unvisited nodes, typically in a *breadth-first* manner. Newer nodes are added subsequently to the community during the traversal. The traversal techniques and the criteria for adding nodes to a community are discussed in detail below. Almost all the existing community identification algorithms are agglomerative techniques.

5.3 Existing methodologies

Community identification algorithms are very sparse in the literature and are mainly greedy heuristics [12, 135]. They try to obtain an induced subgraph that has a relatively maximum edge density within the nuclei N_C and among the nodes of B_C, and minimal external connection from the boundary B_C to the remaining graph. In this section we discuss the metrics that measure the strength of any identified community and act as stopping criterion for the algorithms. Any community identification algorithm takes an input node (or a set of nodes) and concentrates on building the set *C* with nodes from the boundary set B_C iteratively. Eventually the nucleus of the community is strengthened with nodes that yield maximum density. The input node from which the procedure begins is commonly referred to as a *seed vertex.* Though the procedures are similar, the metric that determines the resulting community differs. The metrics defined in this section are with respect to the induced subgraph $C(V_C, E_C)$ for any given input graph *G* (*V*, *E*).

5.3.1 *Connection density ratio*

Chen *et al*. [40] described an algorithm to identify communities based on maximizing the *connection density* ratio. The connection density ratio $\delta(G)$ (in short δ) is the ratio of the internal and external density and is given by,

$$\delta = \frac{\delta_I}{\delta_E}, \qquad (18)$$

where, δ_I and δ_E of the community C are defined as follows. The internal density (δ_I) is the ratio of the number of edges within C to the number of nodes in C.

$$\delta_I = \frac{|E_C|}{|V_C|} \qquad (19)$$

The external density (δ_E) is the ratio of the edges connecting nodes outside the community to the number of boundary nodes.

$$\delta_E = \frac{\sum_i |E(i, j)|}{|B_C|} \quad i \in V_C, j \in (V' - V_C). \qquad (20)$$

Using this edge-density measure, the algorithm extracts the community around a given node in two phases. In the first phase, any adjacent node whose inclusion increases the value of δ is chosen and added to the community. This continues till no more nodes can be added to the community. The second phase examines subsets of nodes in the reduced graph and chooses the one with the maximum δ. Using this average density measure, the algorithm extracts the community around a given node by iteratively adding and removing neighboring nodes and finding the set of nodes which gives a maximum value of L.

Algorithm 1: Community identification algorithm using connection density [CD]

Input:
G (*V*, *E*) // Graph with vertex set V and edge set E
s // Seed vertex

Output:
C // Set of nodes in the Community

Procedure:

1. $C \leftarrow \{s\}$
2. $A \leftarrow N[C]$ // Closed neighborhood of C
3. //*Addition Phase*
4. **do**
5. compute δ_I , δ_E and δ // Compute the density values
6. **foreach** $u \in A$ **do**
7. $C \leftarrow C \cup \{u\}$ // Add a new node to C from A
8. compute δ_I' , δ_E' and δ' // Compute new density values
9. $C \leftarrow C/\{u\}$
10. **end for**
11. $u \leftarrow \boldsymbol{MAX}(\delta')$ // Find the node that yields
12. **if** ($\delta_I' > \delta_I$) **then** // max. increase in δ
13. $C \leftarrow C \cup \{u\}$ // Include nodes that increase δ_I
14. **end if**
15. **while** ($\delta' > \delta$)
16. $tempC \leftarrow C$ // Duplicate the community nodes
17. $C \leftarrow \{s\}$ // Initialize the community again
18. //*Examination Phase*
19. **foreach** $u \in tempC$ **do**
20. $C \leftarrow C \cup \{u\}$ // Include node that increases δ_I
21. compute δ_I' , δ_E' and δ' // and decreases δ_E
22. **if** ($\delta_I' > \delta_I$ && $\delta_E' < \delta_E$) **then**
23. $C \leftarrow C \cup \{u\}i$
24. **end if**
25. **end for**
26. **return** *C*

5.3.2 *Local modularity*

Clauset [43] defined a community based on the density of the boundary edges. His definition included the boundary adjacency matrix B :

$$B_{ij} = \begin{cases} 1 & \textit{if nodes } i \ \& \ j \textit{ are connected and either vertex is in } B_C \\ 0 & \textit{otherwise} \end{cases}$$

He also proposed the local modularity Q_l and defined it as the ratio of the number of edges the boundary vertices (B_C) share with the nuclei (N_C) to the total number of boundary edges. The algorithm iteratively maximizes the value of Q_l by adding new nodes, till a given number of nodes are obtained in the community.

The local modularity is given by

$$Q_l = \frac{\sum_{ij}(B_{ij} * p(i,j))}{\sum_{ij} B_{ij}}. \qquad (21)$$

Here $p(i,j)$ is 1 when either $v_i \in B_C$ and $v_j \in N_C$ or vice versa and is 0 otherwise. Note that the global definition of modularity Q was given by Newman and Girvan [117]. Knowing the number of expected nodes in the community or setting an upper limit on the number of community nodes prior to beginning the algorithm is not an efficient solution in case of complex networks.

Algorithm 2: Maximization of local modularity to identify community [LM]

Input:
$G(V, E)$ // Graph with vertex set V and edge set E
s // Seed vertex
k // Number of vertices to be included in the Community

Output:
C // Set of nodes in the Community

Procedure:

1. $C \leftarrow \{s\}$
2. $A \leftarrow N[C]$ // Closed neighborhood of C
3. $i \leftarrow 0$
4. $Q_i \leftarrow 0$
5. **while** $(|C| < k)$ **do** // Accumulate k nodes in C
6. $i \leftarrow i + 1$
7. **foreach** $u \in A$ **do**
8. $C \leftarrow C \cup \{u\}$ // Add a new node to C from A
9. compute Q_i // Compute new Q value
10. compute $\Delta Q_i \leftarrow (Q_i - Q_{i-1})$
11. $C \leftarrow C/\{u\}$ // Remove the newly added node
12. **end for**
13. $u \leftarrow \boldsymbol{MAX}(\Delta Q_i)$ // Add to C, the node that yields
14. $C \leftarrow C \cup \{u\}$ // max. increase in Q_i
15. compute Q_i // Compute new Q value
16. $A \leftarrow A/\{u\}$
17. $A \leftarrow A \cup N[u]$ // Update the neighbor set of C
18. **end while**
19. **return** C

5.3.3 *Subgraph modularity*

Luo and Wang [103, 104], define local modularity Q_l for a community in terms of the ratio of the number of edges within and outside the induced subgraph.

$$Q_l = \frac{|E_C|}{|\{E'(i, j) \mid i \in V_C, j \notin V_C\}|} \quad (22)$$

They refer to this term as *subgraph modularity*. Their algorithm begins with the seed node and after each iteration adds the node that increases the value of Q_l. The new nodes to be added are chosen from the neighbor list (which gets updated after each addition). The list is also sorted in non-decreasing order based on the degree. Iterating through the list of neighbor nodes twice to find a better set of nodes yields a strong community, but is an expensive operation in large graphs. Moreover the sorting performed to obtain the nodes with maximum degree adds to the computational cost.

Algorithm 3: Maximization of Subgraph modularity to identify community [SM]

Input:
G (*V*, *E*) // Graph with vertex set V and edge set E
s // Seed vertex

Output:
C // Set of nodes in the Community

Procedure:

```
1.  C ← {s}
2.  compute Q_l                          // Compute modularity Q_l for the
3.  do                                   //      initial community
4.      A ← N[C]                         // Closed neighborhood of C
5.      SORT(A)                          // Sort A based on node degree
6.      foreach u ∈ A do                 //      in descending order
7.          compute ΔQ_l                 // Change in Q_l on adding u
8.          if ΔQ_l > 0) then            // Add node to C if Q_l increases
9.              C ← C ∪ {u}
10.         end if
11.     end for
12.     foreach u ∈ C do
13.         C ← C/{u}                    // Change in Q_l on removing
14.         compute ΔQ_l                 //      u from C
15.         if ΔQ_l > 0) then            // Remove node if it increases Q_l
16.             C ← C/{u}                // If C/{u} does not disconnect C
17.         end if
18.     end for
19.     A' ← N[C]                        // Update the new neighbor set
20. while (A' > A)                       // Update the community till no
21. compute Q_l                          //      new nodes can be added
22. if Q_l > 1 and s ∈ C                 // Accept C if Q_l > 1 and
23.     return C                         //      s belongs to C
```

5.3.4 *l-shell spreading using emerging edges*

Bagrow and Bollt proposed an algorithm based on the ratio of *emerging edges* [12]. The number of emerging edges of a node v_i is the out-degree of the node d_i^{out}. When the nodes of the graph are explored breadth-first with the seed as the root, the *total emerging degree* (D_j) is the sum of the out-degree of the nodes at level *j*.

$D_j = \sum_i d_i^{out}$, where node v_i is at level *j* from the root.

Their algorithm iteratively computes the change in D_j and the stopping criterion is given by an input value α. The value of α is predetermined based on the degree distribution of the input graph and it varies from one network to the other. The change in D_j at a level *j* is given by

$$\Delta = \frac{D_j}{D_{j-1}}, \quad j > 0 \qquad (23)$$

and $D_0 = d_u$, where u is the seed node. Predetermining the value of α based on the degree distribution of the input graph is a tedious pre-computation step and will not be optimal in case of large complex networks.

Algorithm 4: *l*-shell spreading algorithm [LS]

Input:
$G(V, E)$ // Graph with vertex set V and edge set E
s // Seed vertex
α // Threshold value used as stopping criterion

Output:
C // Set of nodes in the Community

Procedure:

1. $C \leftarrow \{s$
2. $A \leftarrow N[C]$ // Closed neighborhood of C
3. $D_0 \leftarrow 1$
4. $D_1 \leftarrow d_s$ // Initialize emerging degree of C as degree of s
5. $i \leftarrow 1$
6. $\Delta D_i \leftarrow {}^{D_1}/_{D_0}$ // Initial change in total emerging degree
7. **while** $(\Delta D_i > \alpha)$ **do** // Accumulate nodes to C till threshold value
8. $i \leftarrow i + 1$
9. $C \leftarrow C \cup A$ // Add all neighbors to the current community
10. $A \leftarrow N[C]$ // Update the neighbor set
11. compute D_i // Compute emerging degree of the current C
12. $\Delta D_i \leftarrow {}^{D_i}/_{D_{i-1}}$ // Calculate the change in total emerging degree
13. **end while**
14. **return** C

5.3.5 *Relative edge density metric*

Schaeffer [142] presented a metric that considers not only the density of edges within the community but also the proportion of the edges within and outside. The internal density (δ_I) and relative density (δ_R) values given by Schaeffer are as follows:

$$\delta_I = \frac{2|E_C|}{|V_C|(|V_C|-1)} \quad (24)$$

This value gives the ratio of the number of edges in the community to the total number of edges possible with the given number of nodes. Let E_g denote the set of edges that connect the community nodes (V_C) with the external nodes.

$$E_g = \{E(i,j) | \, i \in V_C, j \notin V_C\} \quad (25)$$

The relative density of the induced subgraph is given by

$$\delta_R = \frac{|E_C|}{|E_C|+|E_g|} \quad (26)$$

$$f = \delta_I * \delta_R \quad (27)$$

The algorithm performs a local search using simulated annealing technique and the subgraph with the maximum value of f is the final community.

Algorithm 5: Relative edge density [RD]

Input:
G (*V*, *E*) // Graph with vertex set V and edge set E
s // Seed vertex

Output:
C // Set of nodes in the Community

Procedure:

1. $C \leftarrow \{s\}$
2. $A \leftarrow N[C]$ // Closed neighborhood of C
3. **do**
4. compute δ_I , δ_R and δ
5. $Choose\ case\ 1\ or\ case\ 2\ Uniformly\ Randomly$
6. **Case 1:**
7. **{**
8. $u \leftarrow\ Choose\ a\ random\ node\ from\ A$
9. $C \leftarrow C \cup \{u\}$ // Add a randomly chosen
10. compute δ_I' , δ_R' and δ' // node and check if it
11. **if** $(\delta' < \delta)$ **then** // increases the density
12. $C \leftarrow C/\{u\}$
13. **end if**
14. }
15. **Case 2:**
16. {
17. $u \leftarrow\ Choose\ a\ random\ node\ from\ C$-{s}
18. $C \leftarrow C/\{u\}$ // Remove an existing node
19. compute δ_I' , δ_R' and δ' // and check if it
20. **if** $(\delta' < \delta)$ **then** // increases the density
21. $C \leftarrow C \cup \{u\}$
22. **end if**
23. }
24. $A \leftarrow N[C]$ // Update the neighbor set
24. **while** $(\delta' > \delta)$
25. **return** *C*

5.4 Improved relative edge density metric

The definition of a community or the measure that determines the strength of a community should be based on the number of edges in the induced subgraph [158]. We adopt such a definition, initially proposed by Schaeffer [142], to consider both the density of edges within the community and the relative ratio of the edges within and outside. A greedy algorithm based on the maximization of this improved parameter f is discussed in the next section. The number of internal edges and external edges from Schaffer's definition [142] are given by

$$d_{int} = |E_C| \quad (28)$$

$$d_{ext} = \sum_{i,j} E(i,j), \ \ i \in V_C and\ j \in V \quad (29)$$

The internal density and relative density values are as follows:

Internal density,

$$\delta_I \ \leftarrow \frac{|E_C|}{\left(\frac{|V_C|(|V_C|-1)}{2}\right)} \quad (30)$$

Relative density,

$$\delta_R \ \leftarrow \ \frac{d_{int}}{d_{int} + \ d_{ext}} \quad (31)$$

The product of these two parameters is proportional to the strength of the community,

$$f \leftarrow \ \delta_I \ * \ \delta_R \quad (32)$$

The definition of d_{ext} reveals that the count does not take into account edges between the nodes adjacent to the induced subgraph.

So a modified definition to the external edge count to include these edges is described.

Let $\partial(g)$ denote the set of all nodes adjacent to the induced subgraph g.

$$\partial(g) = \{ j \in V | E(i,j), where\ i \in V_C, j \notin V' \}$$

So the new definition of the sum of external edges is

$$d_{ext} = |E(i,j)| + |E(k,l), \quad i \in V_C, j \in V\ and\ k,l \in \partial(g)$$

Consider the following example (Figure 13).

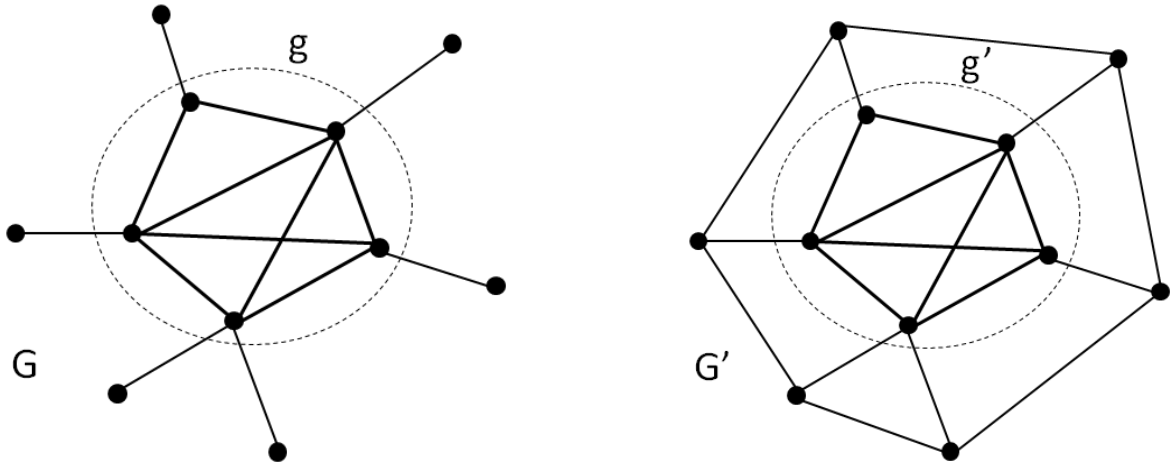

Figure 13: Induced subgraphs with different sparseness

Here, for g, d_{int} = 8 and d_{ext} = 6. Similarly the values for g', are d_{int} = 8 and d_{ext} = 6. The presence of edges between the nodes outside g' does not impact the value of the number of external edges. Therefore, the relative density value would be the same for both the induced subgraphs (g and g'). But the subgraph g is stronger compared to g' since it is a part of a relatively less sparse graph. The modified definition of external degree would count these edges that account for the relative density of the subgraph. Therefore, the value of d_{ext}, according to the new definition would be 6 and 12 for g and g' respectively.

5.5 Community identification based on improved relative density

A greedy algorithm based on maximizing the value of the metric '*f*' is presented and discussed in this section. The metric *f* is defined from the above mentioned inference - the product of internal and relative density. From the definition it is evident that the induced subgraph, that includes the seed vertex, with the maximal value of internal and relative densities would be the ideal community. Therefore, we traverse locally from the seed vertex in a breadth-first manner accumulating nodes that increase the value of *f*. One of the pitfalls of the existing methodologies in identifying a community is the inclusion of outlier nodes. We address this issue in our algorithm which is discussed in detail below.

The seed vertex and its adjacent neighbors form the initial community. The nodes adjacent to this initial community form the neighbor-list (*Queue*). This list is frequently updated depending on the inclusion or removal of nodes in the community during the iteration. The core module of the algorithm is the alternation between two steps – (*i*) addition phase and (*ii*) deletion phase.

In the *addition phase* each node from the queue are added one-at-a-time and the change in density value is checked. We add a node to the current community only if it increases the value of *f*. The new set of community nodes thus obtained (the community maybe unchanged after addition phase, if no new node increases the value of '*f*') is the input to the deletion phase. In the deletion phase, we remove one node at a time and compute the change in *f* value. If there is an increase, clearly, that node does not belong to the community. If the removal of the input seed node increases the density value then it does not belong to any community. The output in that case would be an empty set. The neighbor-list is updated at this point to include nodes adjacent

to the newly added nodes. The alternative addition and deletion continues until no new vertices can be added to increase the value of the density. The pseudocode of this technique is expressed in Algorithm 6.

The greedy algorithm based on the improved relative density metric uses the addition and deletion strategy similar to the one employed in Luo *et al*. algorithm [104] but in a more effective way. The metric used in their algorithm just measures the ratio of the internal and external edges but discards the significance of the number of nodes in the induced subgraph. Since we are interested in dense subgraphs of significant size it is essential to keep into consideration the number of nodes added to the community. The addition phase of our algorithm ensures the inclusion of nodes that contribute even slightly to the strength of the community. The deletion phase takes care of eliminating outlier nodes that is of little importance to the community.

Suppose the final resulting community consists of k' nodes and the average degree of the subgraph is $\bar{d}$. The average time complexity of the algorithm is $O(k'^2\bar{d})$. Since in each iteration $k'\bar{d}$ adjacent nodes of each of the k' nodes in the community needs to be investigated for inclusion or exclusion from the induced subgraph.

Algorithm 6: Community identification based on improved relative density

Input:
G (*V*, *E*) // Input graph with vertex set *V* and edge set *E*
u // Seed vertex

Output:
C // Set of vertices that form the Community

```
1.  Procedure Initialize ( )
2.  {
3.      C ← {u}
4.      Queue ← ∅                          // To keep track of visited vertices
5.      foreach w ∈ Adj(u)
6.      {
7.          enqueue (w, Queue)             // Insert node w to the Queue
8.          C ← C + {w}
9.      }
10.     f ← computeDensity ( )             // Compute the value of f
11.     do
12.     {
13.         CurrentSize ← |C|              // Store the current size of the community
14.         additionPhase ( )
15.         deletionPhase ( )
16.     }while (CurrentSize ≠ |C|)         // Repeat till there is no change in size
17. }
18.
19. Procedure additionPhase ( )            // Add new node and compare change in value
20. {
21.     u ← dequeue(Queue)                 // Obtain the first element of the Q
22.     foreach w ∈ Adj(u)
23.     {
24.         enqueue (w, Queue)
25.         C ← C + {w}
26.         f' ← computeDensity ( )
27.         if (f' < f) then               // Density value does not increase
28.             C ← C − {w}
29.     else                               // Retain the new value
30.         f ← f'
31.     }
32. }
```

```
33.  Procedure deletionPhase ( )          // Delete existing node and compare change in value
34.  {
35.      foreach  u ∈ C
36.      {
37.          C ← C − {u}
38.          f′ ← computeDensity ( )
39.          if (f′ < f) then                    // Density value does not increase
40.              C ← C + {u}
41.          f ← computeDensity ( )
42.      else                                    // Retain the new value
43.          f ← f′
44.          deletefromQ(u)
45.      }
46.  }
```

5.5.1 *Experimental observation*

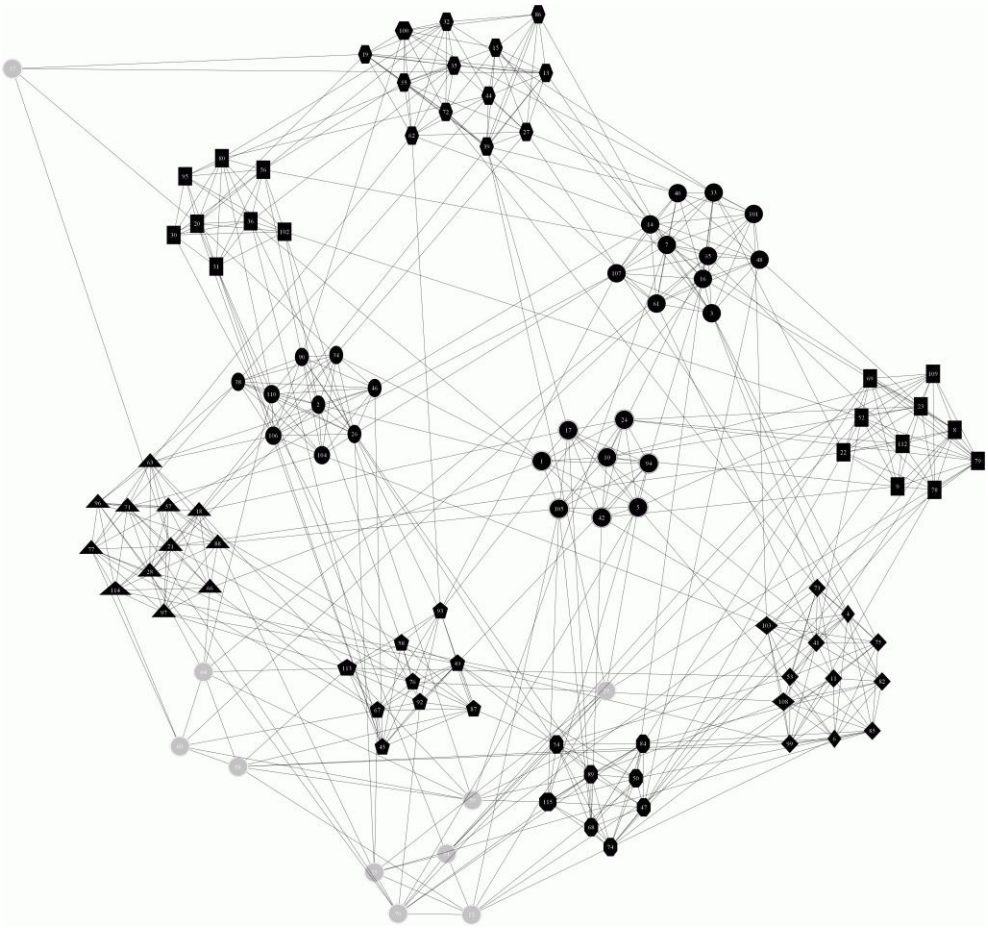

Figure 14: NCAA football network of fall 2000 season. The ten communities identified correspond to the ten conferences.

The algorithm was implemented and tested on the Girvan and Newman synthetic graph (discussed in detail in Chapter 7) consisting of 128 nodes, 1024 edges and four communities. One node from each of the communities was used as the seed vertex and the algorithm identified the corresponding community accurately. The implementation was also tested with different

nodes from the same community and on each instance it yielded the community corresponding to the input seed node. However the most interesting observation was made on the communities obtained in the American football network compiled by Girvan and Newman [72]. The network formed by the NCAA Division IA colleges during regular season fall 2000 has been used as a benchmark to test several community detection algorithms in the literature and was believed to have 11 communities corresponding to the 11 conferences in the league. But our algorithm identified only ten communities which had a relatively dense sub structure as shown in Figure 14. The ten communities were obtained by using input seed nodes from different conferences, but the node from the eleventh conference did not yield any community. Later it was acknowledged by Evans [59] that the 2000 football season indeed had only ten conferences and more games were scheduled among them than against other conference teams. The algorithm was also tested on other real-world networks such as the Zachary karate club, Bottlenose dolphins, etc., with known community structures.

CHAPTER 6: COMMUNITY IDENTIFICATION ALGORITHM BASED ON AVERAGE DEGREE

In this section a novel divisive algorithm to identify communities in complex networks based on maximizing average degree is presented. Even though the improved density based definition discussed in the previous chapter is effective in retrieving dense communities, computing internal and external densities at each step could prove costly in large communities. Moreover the existing techniques are all agglomerative (accumulate nodes to the community during each iteration) and hence can be computationally expensive for large networks. A discussion on the definition based on average degree, the NP-completeness of the community identification and the algorithm are as follows.

6.1 Definition

Given a graph $G(V,E)$, the average degree of an induced subgraph $g(V',E')$ is defined as the ratio of twice the number of edges in the subgraph to the total number of nodes in g. Average degree of the subgraph g is given by

$$\bar{d}_{V'} = \frac{2*|E'|}{|V'|} \qquad (33)$$

Note that the sum of the degrees of all nodes in a graph is twice the number of edges in the graph [46]. An example graph with average degree of its subgraph is shown in Figure 15. There are five nodes and eight edges in the subgraph. The sum of the degree of the nodes is 16 and so the average degree is $16/5 = 3.2$.

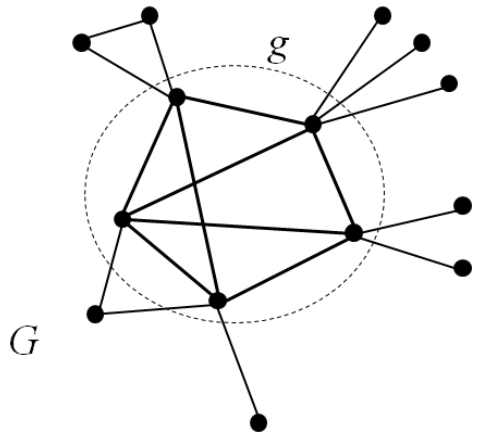

Figure 15: Average degree of the subgraph g is 3.2.

The subgraph $g(V',E')$ forms a community if the average degree of g ($\bar{d}_{V'}$) is greater than the average degree of any other subset of vertices within the neighborhood of V'. Since complex networks are large, we restrict ourselves to considering only the set of vertices within a neighborhood at a distance *k*. Therefore, if $\Gamma(k,u)$ denotes the set of vertices within a neighborhood *k* from vertex *u*, then $V' \subseteq \Gamma(k,u)$ forms a community if

$$\forall V'' \subseteq \Gamma(k,u), \qquad \bar{d}_{V''} < \bar{d}_{V'} \qquad (34)$$

Since, in a community identification algorithm we begin with a seed vertex $u \in V$, defining a community by considering vertices only within *k*-neighborhood is a realistic measure. Next, a divisive algorithm based on maximizing the average degree of the induced subgraph within *k*-neighborhood is elucidated. The NP-completeness of the problem is discussed prior to that.

6.2 NP-Completeness

The above definition of a community depicts the dense subgraph in large sparse graph more precisely than the existing definitions (elaborated in Chapter 3). But, identifying a subgraph with maximum average degree in the given graph would require a comparison of all the subsets of nodes leading to an exponential solution. The NP-completeness of community identification can be proved by *restriction* method given by Garey and Johnson [69]. An NP-completeness proof by restriction for a given problem Π consists of showing that Π contains a known NP-complete problem $\Pi' \in NP$ as a special case. Identifying an induced connected subgraph with a specific property is a proven NP-complete problem (GT22: [69]). One such NP-complete problem is the identification of cliques, where the property of the induced subgraph is the completeness. The problem of identifying an induced connected subgraph with maximum average degree restricts to this NP-complete problem, since identifying cliques would be a special case of our community identification.

6.3 Community identification based on maximizing average degree

The definition of a community clearly indicates that we are interested in identifying subsets of vertices with maximal average degree in a given graph. Given a large graph and a seed node *u* we identify a set of vertices that *u* belongs to satisfying the above criteria. Since we explore only a restricted neighborhood of *u*, there is one other parameter *k,* required as input to the algorithm. In other words, the parameter *k* specifies the distance till which we explore the given graph from the seed. Typically, the value of *k* is a positive integer greater than two. When *k* value is one or

two, we have just the seed node and/or few of its neighbors (neighbors of neighbors), which is insufficient to identify a community.

As a first step of the algorithm we perform a *breadth-first search* [46] on the input graph starting at node *u*. Once we obtain the subset of vertices $\Gamma(k,u)$ by the breadth-first technique, our algorithm iteratively removes nodes that do not belong to the community. Let *C* denote this initial set of vertices $\Gamma(k,u)$. The average degree of the subgraph [$\bar{d}_C$] is calculated. Now, we remove the nodes with minimum degree and calculate the new average degree [$\bar{d}_H$]. We compare the two average degree values and if $\bar{d}_H$ is greater than $\bar{d}_C$ then we store the new set of vertices in *C*. It is to be noted that the degree of the nodes in *C* needs to be updated on the removal of minimum degree nodes. Again, with the new set of vertices, we remove the nodes with the least degree and re-compute the average degree $\bar{d}_H$. The iterative removal and average degree computation continues till $\bar{d}_H$ is greater than $\bar{d}_C$. If $\bar{d}_H$ becomes less than $\bar{d}_C$, we stop the iteration and return the set of nodes *C* as the community. If a given seed does not belong to any community, it will be removed in one of the iterations of the algorithm (since the node would not contribute to a higher average density of the resulting set of nodes). A detailed description of the notation and steps involved in the algorithm are given by Algorithm 7.

Algorithm 7: Community identification based on Maximizing Average degree [AD]

Input:

$G(V, E)$	Graph with vertex set V and edge set E
u	Seed vertex
k	Depth of BFS exploration from u

Output:

$C(V_C, E_C)$	Set of vertices and edges that form the Community

```
1   Procedure Initialize ( )
2   {
3       H ← BFS(G, u, k)          // Function returns the induced subgraph till
4                                 //   depth k from node u as root, in graph G
5   }
6   Procedure find_Community ( )
7   {
8       C ← H                     // Start with the entire subgraph as C
9       d_C ← Compute_Avg_Deg (C) // Find average degree of the subgraph C
10      do
11      {
12          H ← C                 // Previous best Community
13          d_H ← d_C             // Previous best Avg. deg
14          d_min ← min_{v∈V_C} d_v   // Find the smallest degree of the subgraph
15          V' ← {v ∈ V_C | d_v > d_min}   // Include nodes that have degree > d_min
16          V_C ← V'
17          E_C ← {(u, v) | u, v ∈ V_C} ∩ E_C   // Include newly induced edges
18          d_C ← Compute_Avg_Deg (C)   // Find avg. deg. of the modified subgraph
19      }while (d_C > d_H)
20      C ← H                     // Restore best Community
21      return C
22  }
```

6.4 Computational complexity

The time complexity of the algorithm is $O(n'\bar{d})$, where *n'* denotes the number of nodes in the *k*-neighborhood [$n' = \Gamma(k,u)$] and $\bar{d}$ denotes the average degree of the nodes in the subgraph. For each of the nodes (*n'*) in the resulting community, the algorithm needs to check the degree of the adjacent nodes $(n'\bar{d})$. The time complexities of the existing community identification algorithms are mentioned in Table 4. Let *k'* denote the number of nodes in the resulting community for a given input node and $\bar{d}$ denote the average degree of the nodes, then the average case time complexity of the algorithms are as follows:

Table 4: Average case complexity of community identification algorithms

Algorithm	**Complexity**
Clauset's local modularity [LM]	$O(k'^2\bar{d})$
Bagrow & Bollt's *l*-shell [LS]	$O(k'\bar{d})$
Luo *et al.* subgraph modularity [SM]	$O(k'^2\bar{d} + (k'\bar{d})\log(k'\bar{d}))$
Chen *et al.* connection density [CD]	$O(k'^2\bar{d} + k')$
Schaffer's relative density [RD]	$O(k'^2\bar{d})$
Improved relative density	$O(k'^2\bar{d})$
Maximizing average degree	$O(n'\bar{d})$

CHAPTER 7: PERFORMANCE ON SYNTHETIC AND REAL-WORLD NETWORKS

One of the desired characteristic of a community identification algorithm is the accuracy to fetch the set of vertices that form the community irrespective of the input seed. Since an accepted quantitative metric to gauge the quality of a community is yet to be devised in the literature, the algorithm has to be compared and tested against graphs with known community structures. Synthetic and real-world graphs have been used as benchmarks to test the effectiveness of community detection algorithms. We implemented our algorithm using Java in eclipse SDK and the graph visualizations were done using Gephi [20], GLay[154] and Pajek [21].

Regular graphs [e.g., ladder, path, grid] do not posses any dense subgraphs because their edges are uniformly distributed. Hence they are not investigated for presence of communities. Classical random graphs also do not exhibit this property since there is no preferential attachment among nodes to form denser subgraphs [34]. Therefore it is essential to use random graph models that generate scale-free, clustered graphs to test community detection algorithms. Several random graph generators take into account the dynamic addition of nodes and edges and produce graphs similar to real-world complex networks [42, 47, 63, 116, 126, 159]. Similarly, numerous examples of complex networks exist in real-world such as the web graph [52], protein-interaction networks [105] and the Internet [15]. But it is essential that the algorithms are tested on graphs with known community structures, before applying them to real-world large graphs. The execution results of our algorithm on these synthetic graphs and some real-world graphs are discussed below.

7.1 Synthetic graphs

A number of network models have been proposed in the literature to generate graphs replicating the properties of complex networks [34, 94, 169]. In particular, the random graphs generated with locally-dense subgraphs are the ones of interest to test community detection algorithms. Algorithms are tested on these controlled graphs to see if they recognize the already known community structure. The performance comparison of the community identification algorithms on two such synthetic graphs, which are most cited in the literature are discussed.

7.1.1 *GN graphs*

Girvan and Newman described a synthetic graph with 128 nodes divided equally into four communities (GN graph) [117]. The average degree of the graph is equal to 16 and so the graph consists of 1024 edges. Edges are placed independently at random between the vertex pairs with probabilities p_{in} (higher value - for an edge within the same community) and p_{out} (low value - for an edge between two communities). One node from each of the communities was chosen randomly as input to our algorithm and the set of nodes that form the community was correctly identified. The value of k [the BFS depth] was given as three which was sufficient to identify the community. The average degree of the communities is equal to 14.4375. The four communities identified are depicted with different colors in Figure 16.

GN graphs with 256 (GN256) and 512 (GN512) nodes were also used as test graphs to the algorithm. The average degree of each of these graphs was 16 and they consisted of four equal-size communities similar to the 128 node GN graph (GN128). A comparison of the execution results of our algorithm against four other algorithms is shown in Figure 17. All the algorithms

were given the same seed and their execution time to identify the community is calculated in milliseconds. Note that we have not compared the execution time of the algorithm proposed by Bagrow and Bollt because of the aberration in results due to the input parameter α. As evidently seen from the figure, the increase in the number of nodes in the resulting community significantly affects the performance of the existing algorithms. This is a serious bottleneck when applying the algorithms to large real-world complex networks. But the average degree based algorithm (*AD*) takes considerably less time (in fact sub-linear time complexity) and accurately identifies communities irrespective of the size of the graph (or the community).

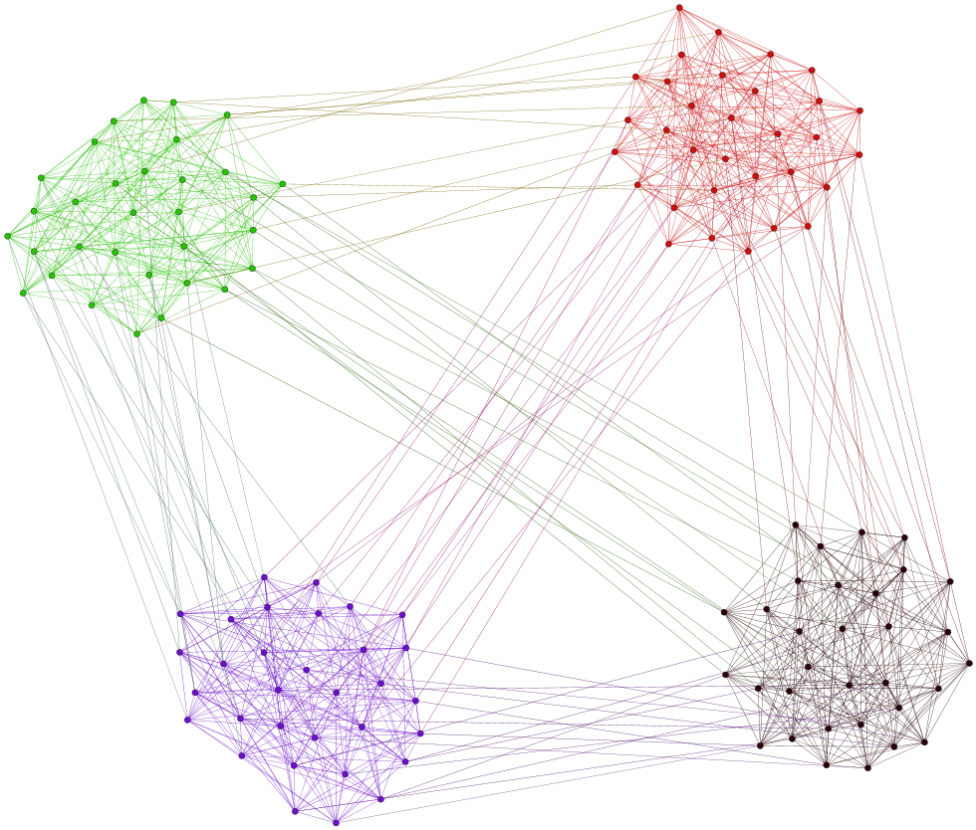

Figure 16: Girvan-Newman synthetic graph (128 nodes, 1024 edges and four equally sized communities).

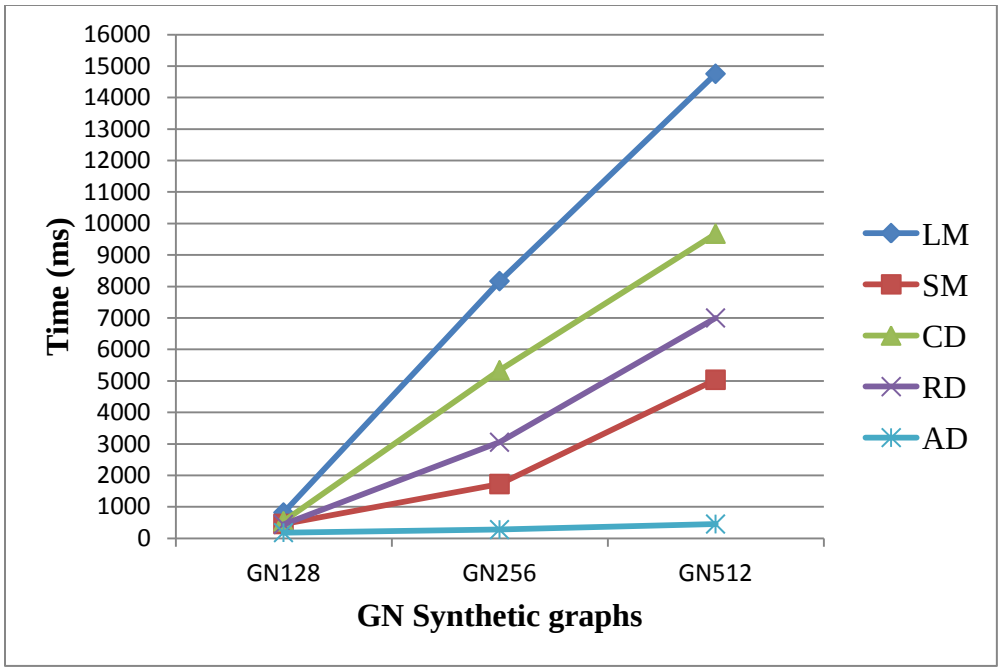

Figure 17: Run time comparison of community identification algorithms

The performance of our community identification algorithm was also tested on GN graphs with varying ratios of inter-community and intra-community edges. We compared the results with two algorithms with the next best run time (inferred from Figure 17) - Luo *et al.*'s algorithm (*SM*) and Schaeffer's algorithm (*RD*). The vertices of the GN graph were divided into four communities of equal size with each vertex consisting of 16 adjacent nodes ($d_i = 16$). The number of inter-community edges was varied from one to nine ($1 \le d_{out} \le 9$). Therefore, a total of nine different GN graphs were generated as test cases. The algorithms were tested using the same seed in each of the test cases and *Jaccard's index* was used to calculate the percentage of nodes correctly identified [132]. If T denotes the target community vertices for a given seed

vertex and C denotes the set of vertices identified for the seed by a community identification algorithm, then Jaccard's index is given by

$$J(T,C) = \frac{|T \cap C|}{|T \cup C|}$$

The value of J ranges from 0 to 1, with 1 indicating a perfect match between the expected and obtained result with no outlier nodes.

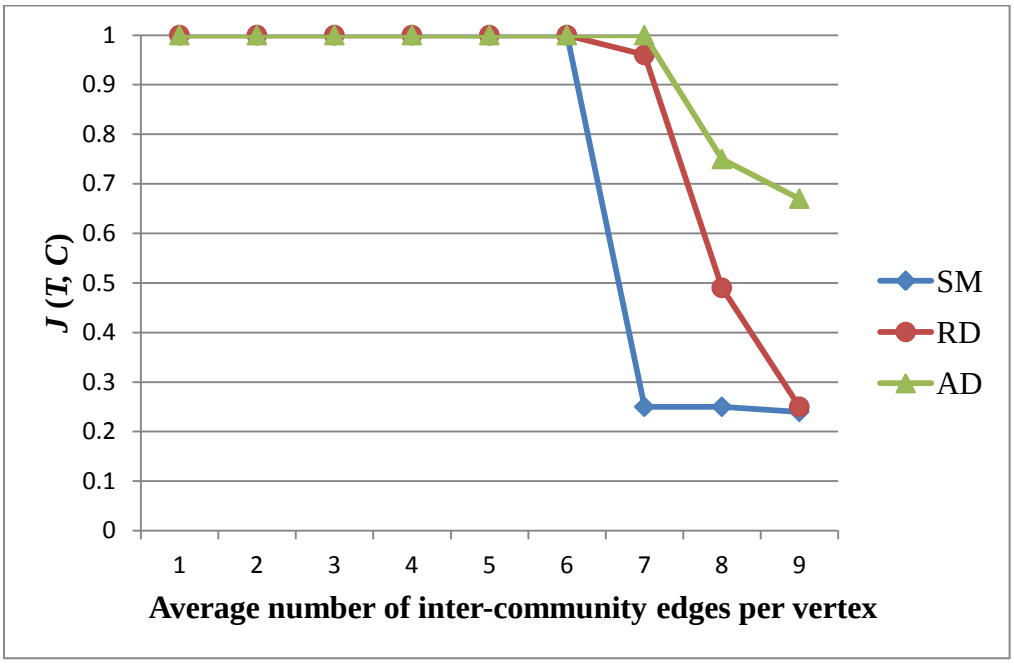

Figure 18: Jaccard index values comparing target set and obtained set of community nodes for three algorithms.

The comparison results are shown in Figure 18. All three algorithms identify the community nodes with no false or true positives until the number of inter-community edges per vertex is less than or equal to 6 [$d_{out} \leq 6$]. But for $d_{out} \geq 7$, *SM* and *RD* algorithms fail to identify all the

nodes in the community a given seed belongs to, whereas our algorithm has $J(T,C)$ equal to one. Even in the case where the number of adjacent nodes within the community is equal to the inter-community degree (i.e., $d_{out} = 8$), the average degree based algorithm identifies 75% of the expected community nodes.

7.1.2 *LFR graphs*

Another model for generating a controlled test graph with communities was proposed by Lancichinetti *et al*. [94, 95]. These graphs are referred as LFR (Lancichinetti, Fortunato and Radicchi) graphs and consist of communities of different sizes unlike GN graphs.

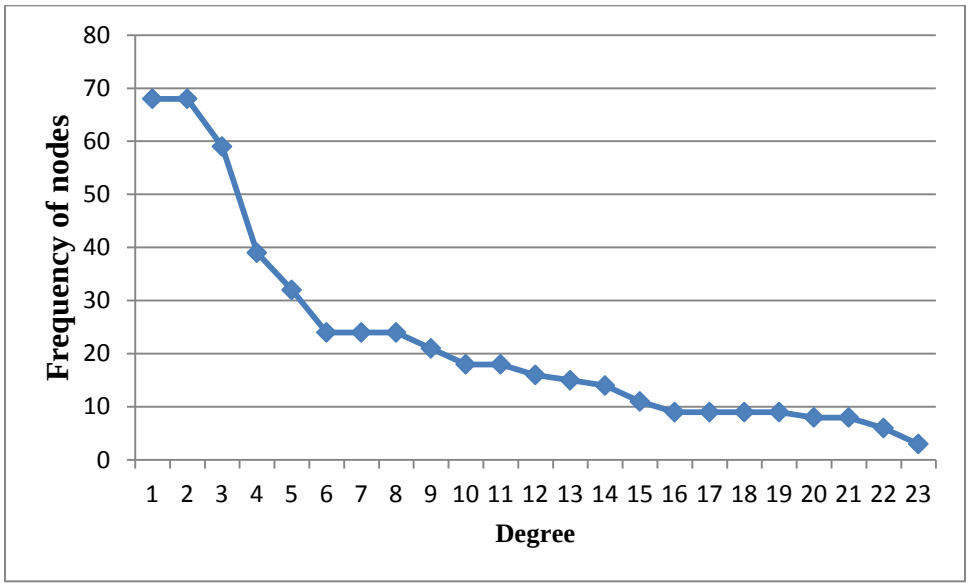

Figure 19: Power-law degree distribution exhibited by LFR graph with 512 nodes.

LFR graphs are better in testing a community algorithm because the communities are of variable sizes and the average degree also varies. We used an LFR graph with 512 nodes and 4171 edges,

with an average degree of 16.293. The degree distribution of the LFR graph is shown in Figure 19 and it clearly exhibits power-law degree distribution discussed in Chapter 2.

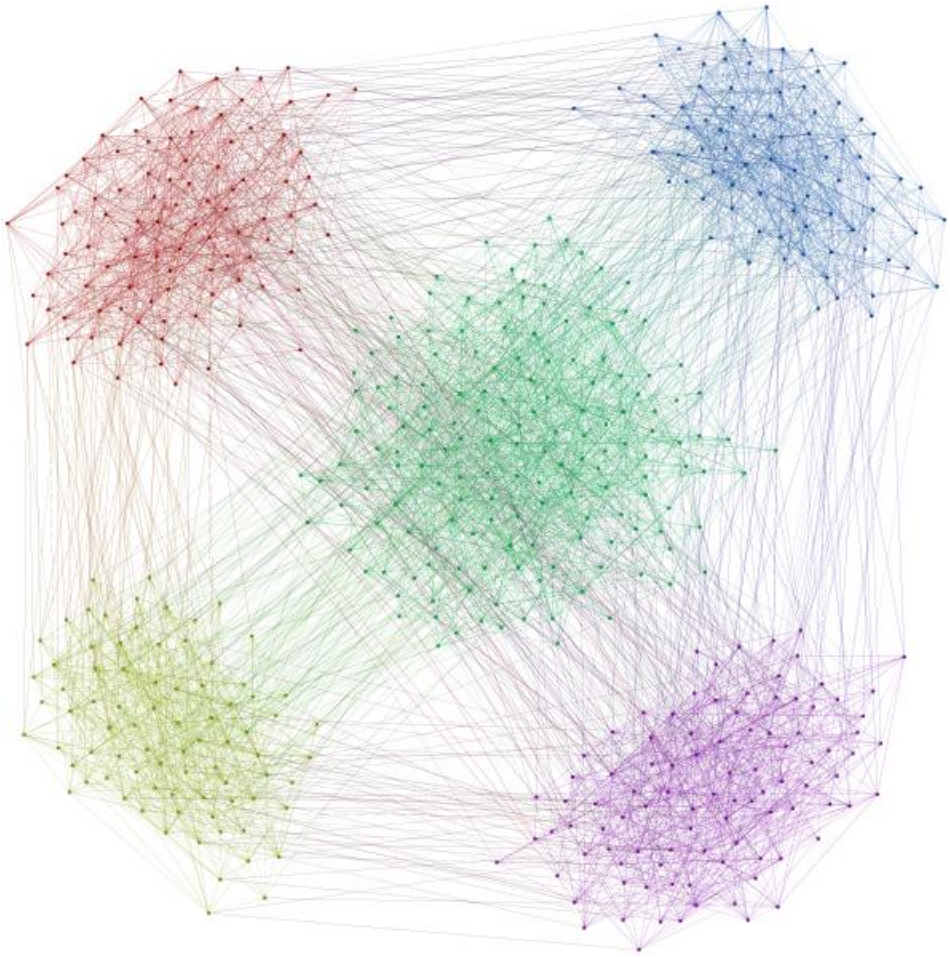

Figure 20: LFR synthetic graph (512 nodes and 4171 edges).

The graph consisted of five communities of varying sizes and is shown in Figure 20. Similar to the GN graph, we selected one node at random from each of the communities and the algorithm identified the corresponding community precisely. Nodes within a distance of four

from the seed node were necessary to identify communities in LFR graphs as against three in case of GN graphs (i.e., k = 4 in the algorithm). The five communities (differentiated using colors) identified by our algorithm are shown in the figure.

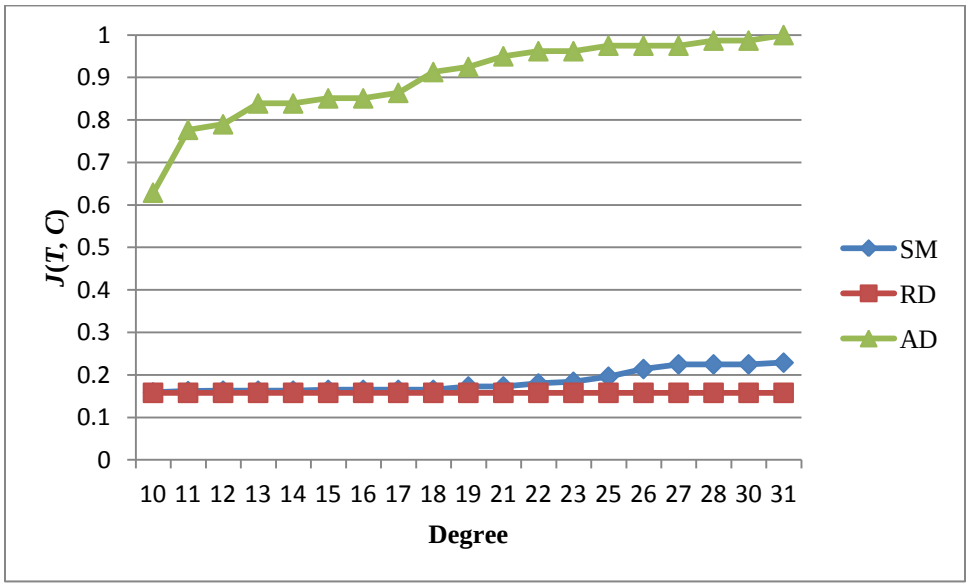

Figure 21: Jaccard's index for communities from seed nodes with increasing degree.

Another interesting observation on community identification algorithms would be the variation in the set of nodes obtained based on the initial seed node. Eventually, we are interested in the same set of nodes that contribute to the community, irrespective of the node from which we begin the identification. This is one of the desired properties of a community identification algorithm and this behavior was tested in the AD algorithm by starting with seed nodes of varying degree (increasing order). One of the five communities (with 81 nodes) was chosen and the degree of the nodes varied from 10 to 31. Figure 21 depicts this comparison for seed nodes from the same community but of varying degrees against SM and RD algorithms. Nodes with

relatively lower degree in the community do not lead to identifying the entire expected set of nodes in the community and hence the lower value of Jaccard's index. The consistency of our algorithm irrespective of the initial seed is evident from the Jaccard's index range (0.65 to 1.0). The other two algorithms did not fare well in LFR graphs as evident from their poor identification of community nodes for a given seed.

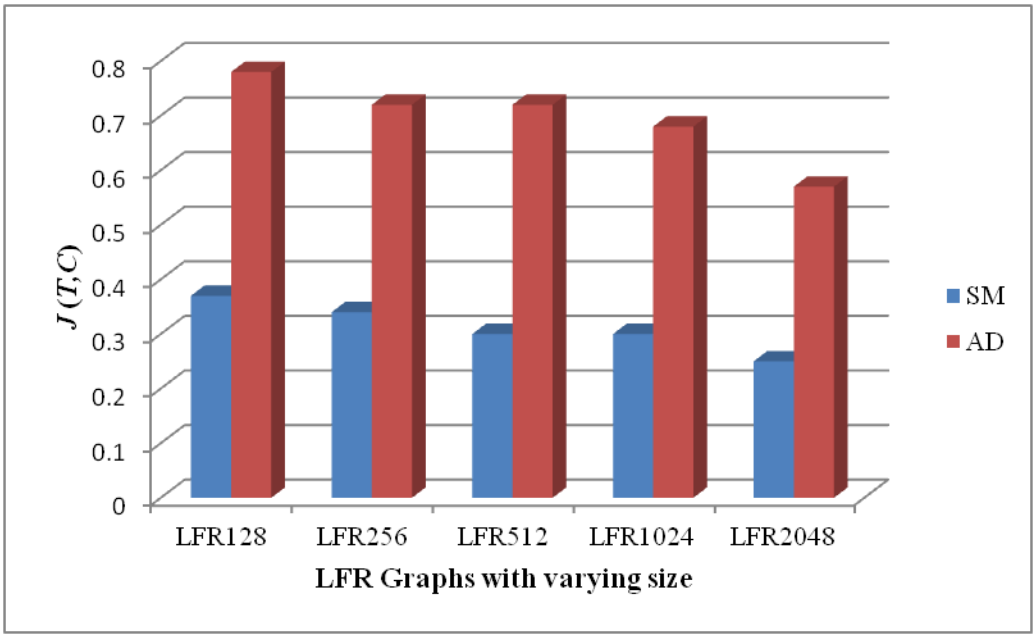

Figure 22: Jaccard's index comparing communities identified in LFR graphs with varying size.

The performance of our algorithm on identifying communities in LFR graphs of increasing magnitude was also tested. The graph sizes chosen were 128, 256, 512, 1024 and 2048. LFR*x* refers to graph with *x* nodes. The number of communities was fixed to five and so the community size also increased with respect to the graph size. Jaccard's index comparing our algorithm with Luo *et al.*'s algorithm on these graphs is shown in Figure 22. Despite the increasing size and decrease in the ratio of intra-community and inter-community edges, our

algorithm identified more than 50% of the expected nodes against only 20% by the subgraph modularity (SM) algorithm. The number of nodes, number of edges and the average degree of the LFR graphs used to test the algorithms are listed in Table 5. Note that Schaeffer's algorithm (RD) did not identify communities in LFR graphs (Figure 21), so we have excluded it from analysis for LFR graphs with increasing size.

Table 5: LFR graphs of varying size and their average degree

Network Type	***n***	***m***	**Avg. degree**
LFR1	128	502	7.84375
LFR2	256	1036	8.09375
LFR3	512	2087	8.15234
LFR4	1024	4145	8.0957
LFR5	2048	8266	8.07227

7.2 Real-world networks

Identifying communities in real-world complex networks tests the effectiveness of a community identification algorithm. As mentioned before, numerous examples of real-world networks have been used as benchmarks to test community detection algorithms in the literature [7, 49, 84, 124, 125, 171]. Some of the real-world benchmark graphs are not very large networks (less than 500 nodes), but are still considered because of their well-defined community structure. For example, Zachary's karate club network [171] is the most cited real-world social network (in community detection literature), but it is a very small network consisting of only 34 nodes. The karate club network and the two communities are shown in Figure 23.

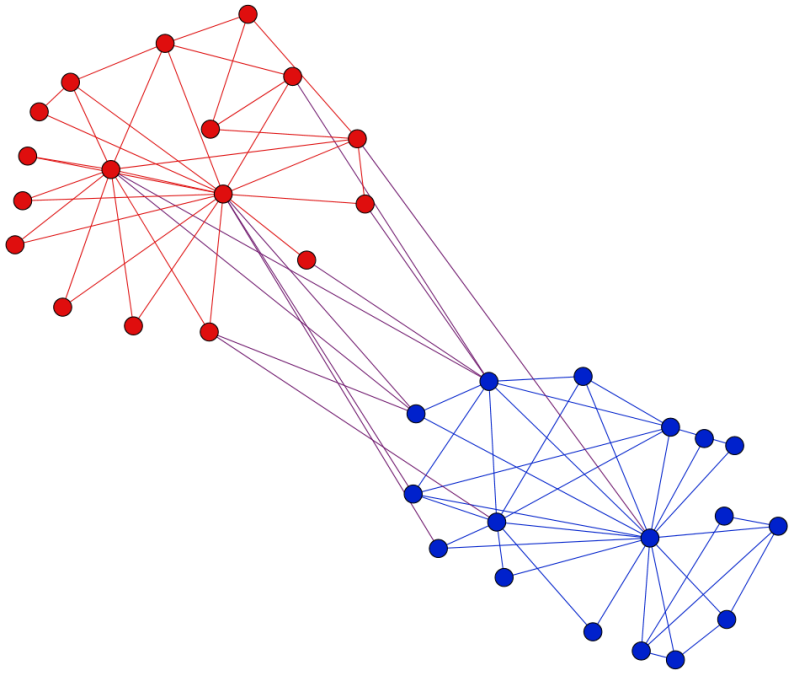

Figure 23: Zachary karate club network (34 nodes and 78 edges)

We tested our algorithm on several known real-world graphs and some of the results are discussed below. Table 6 gives a list of real-world complex networks that we used as input to our algorithm to identify communities. The number of nodes, number of edges, number of communities, and average degree are all summarized in the table. Some of these graphs, their communities and the performance of the algorithm in identifying communities in the real-world networks are discussed in this sub section. The network properties and communities in biological networks are mentioned in a separate sub section. Even though the functional brain network is a biological network, due its abstract nature it is explained in the generic category.

Table 6: Some of the real-world benchmark networks

Network	**Type**	***n***	***m***	$\bar{d}$	N_c	***Ref.***
Zachary Karate Club	Social	34	78	4.6	2	[171]
Bottlenose dolphins	Social	62	159	5.1	2	[106]
Brain Functions	Biological	90	337	7.5	5	[80]
Books on American Politics	Business	105	441	8.4	2	[90]
NCAA Football (2011)	Social	120	674	5.6	11	[1, 2]
Jazz Musicians	Social	198	2742	27.7	2	[73]

7.2.1 Bottlenose dolphins

Lusseau [106, 107] studied the association between bottlenose dolphins in Doubtful Sound, New Zealand. The network was based on social acquaintances (preferred companionships) and each node of the network corresponds to a dolphin. Individuals that were seen together more often than expected by chance have an edge between them in the graph. Lusseau's work focused mainly on the social network properties such as the power-law distribution, clustering and self-organizing phenomena of bottlenose dolphins.

He also observed centers of associations, with adult females being responsible for the hubs. These findings lead to the observance of communities in such a small network and a number of community detection algorithms in the literature explore the intrinsic community nature of this network [11, 38, 117]. Our algorithm identified the core of the two large communities in the dolphin network, ignoring the outlier nodes. These communities correspond to the two groups described by Lusseau *et al*. [107] based on their frequent interaction and association. The two groups of dolphins are denoted by red and blue color nodes (black color nodes represent dolphins that were considered as outliers) in Figure 24.

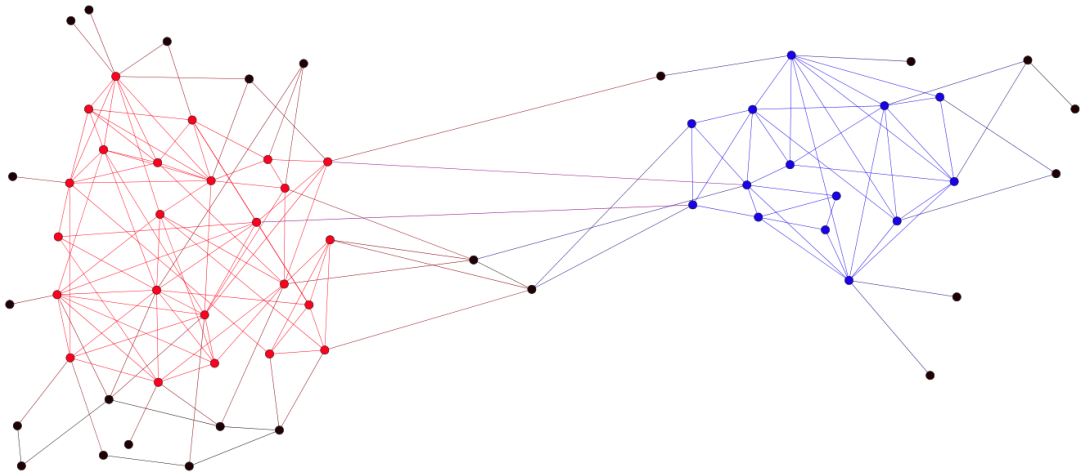

Figure 24: Bottlenose dolphins' social network (62 nodes and 159 edges).

7.2.2 Jazz musicians

The collaboration network of jazz musicians reveals interesting topological and structural properties [73]. Each node corresponds to a jazz band and an edge exists between two nodes if they have at least one musician in common. The network consists of 198 nodes and 2742 edges and its properties has been analyzed in the literature [54, 155, 163]. As in the case with any social network, the jazz musician collaboration network also exhibits communities. The node with the highest degree was chosen as the seed and our algorithm identified the largest dense subgraph of the network as shown in Figure 25 (the community shown in red). The community corresponds to the strong correlation amongst the bands due to their recording locations. Our algorithm identifies the nodes that lie intermediate between the two dense structures of the network to belong to the larger group, which is a desirable result in the real-world scenario.

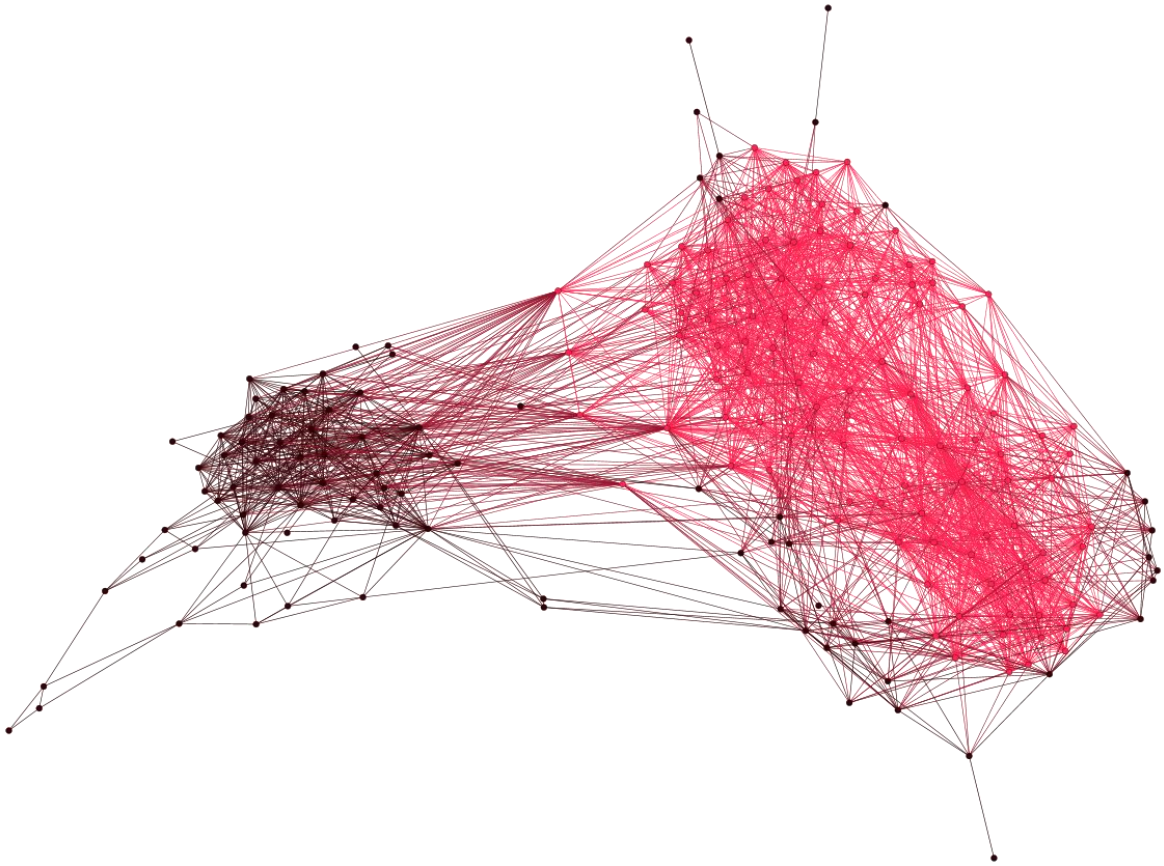

Figure 25: Jazz musicians collaboration network (198 nodes and 2742 edges).

7.2.3 NCAA football

The National Collegiate Athletic Association (NCAA) football network depicts the schedule of football games between Division IA (now known as the Football Bowl Subdivision) colleges. The college teams represent the nodes and an edge between two nodes denotes a regular season game scheduled between the two teams. Girvan and Newman [72] compiled the schedule of the college teams during regular season of Fall 2000 (Figure 14) and it has been used as a benchmark in several community detection algorithms [13, 40, 96, 130]. The number of teams

has been increased from 115 to 120 in 2011 and the football network comprising of these 120 nodes from their 2011 schedule was compiled [1, 2].

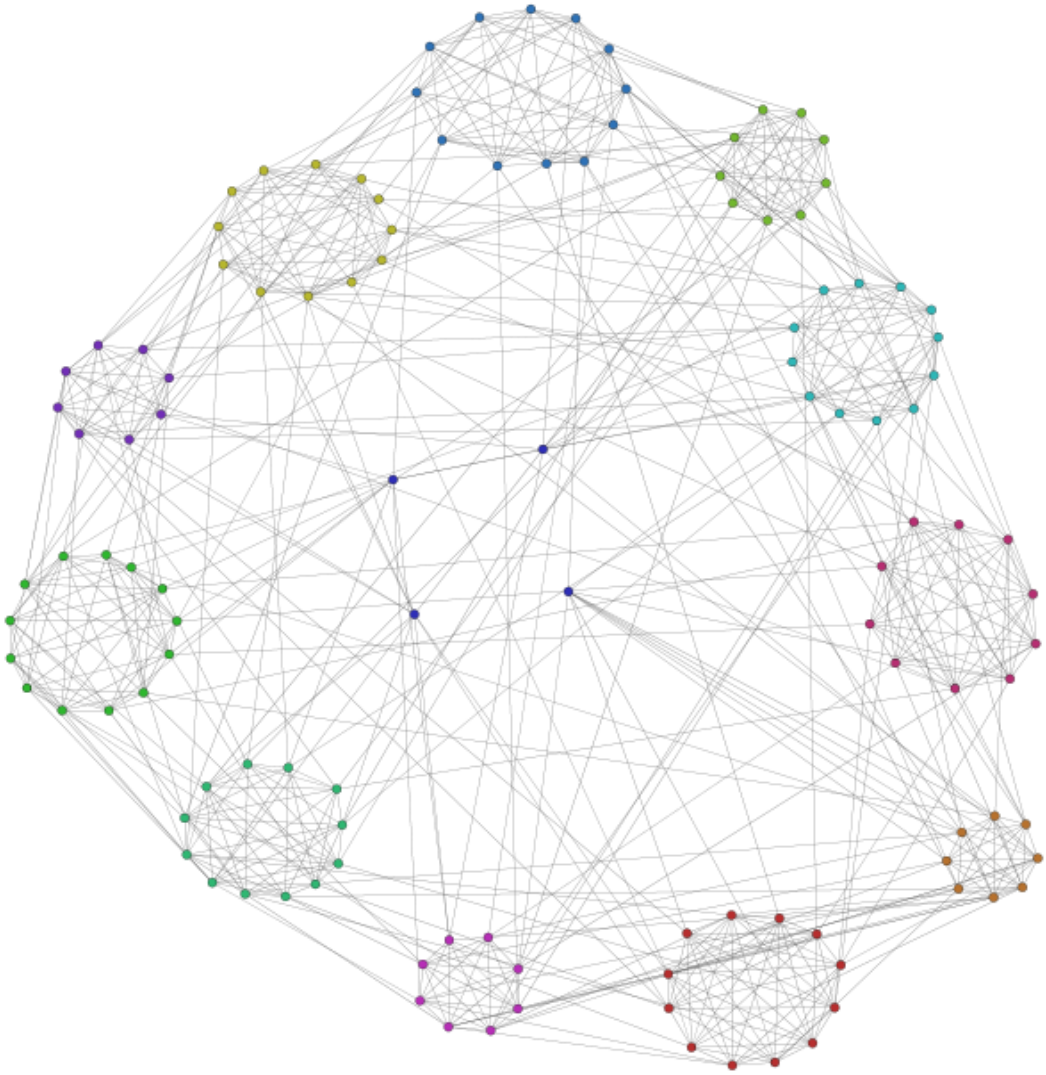

Figure 26: 2011 NCAA Division I FBS football network (n = 120 and m = 674).

Each of the teams belongs to one of the twelve conferences and size of the conferences ranges from 8 to 12 (apart from the *Independents* conference which has only four teams). The

inherent community structure in this network is a result of the conferences because the teams of the same conference play against each other more frequently than they do against teams of other conferences. Therefore, the network comprises of more edges within the conference than outside, thus leading eleven communities (the *Independents* conference does not form a community). Our algorithm distinctly identified the community (in this case the conference) that a given node belongs to and the communities are shown in Figure 26. The eleven communities in the network correspond to the conferences (represented by different colors). The four teams from the Independents conference are placed separately in the middle since they do not form a community.

7.2.4 Brain functions

Analysis on intrinsic human brain activity based on blood oxygen level-dependent (BOLD) and resting-state functional magnetic resonance imaging (R-fMRI) has received considerable attention in the past decade [162]. The reason for this is primarily attributed to the understanding of the brain as a complex network (the *human connectome* [78, 151]). Specifically, the functional network of the brain demonstrates non-trivial topological properties such as small average distance, modularity and highly connected hub regions.

These network properties have also been found to change through aging and in various pathological conditions. The functional brain network, an abstract representation of the brain, comprises of 90 nodes, each of which corresponds to regions in the brain (based on a prior Automated Anatomical Labeling [AAL]) [80]. The temporal correlation between the regions corresponds to the edges among the nodes. This abstract representation of the brain can be

organized as five different communities corresponding to somatosensory (motor and auditory), vision, attention, default-mode and limbic. Given a seed (region), our algorithm returned the corresponding community it belongs to. This helps in identifying the functionality of a region or a sub-region. The five communities are represented by five different colors (Figure 27) and it is evident that they are tightly knit.

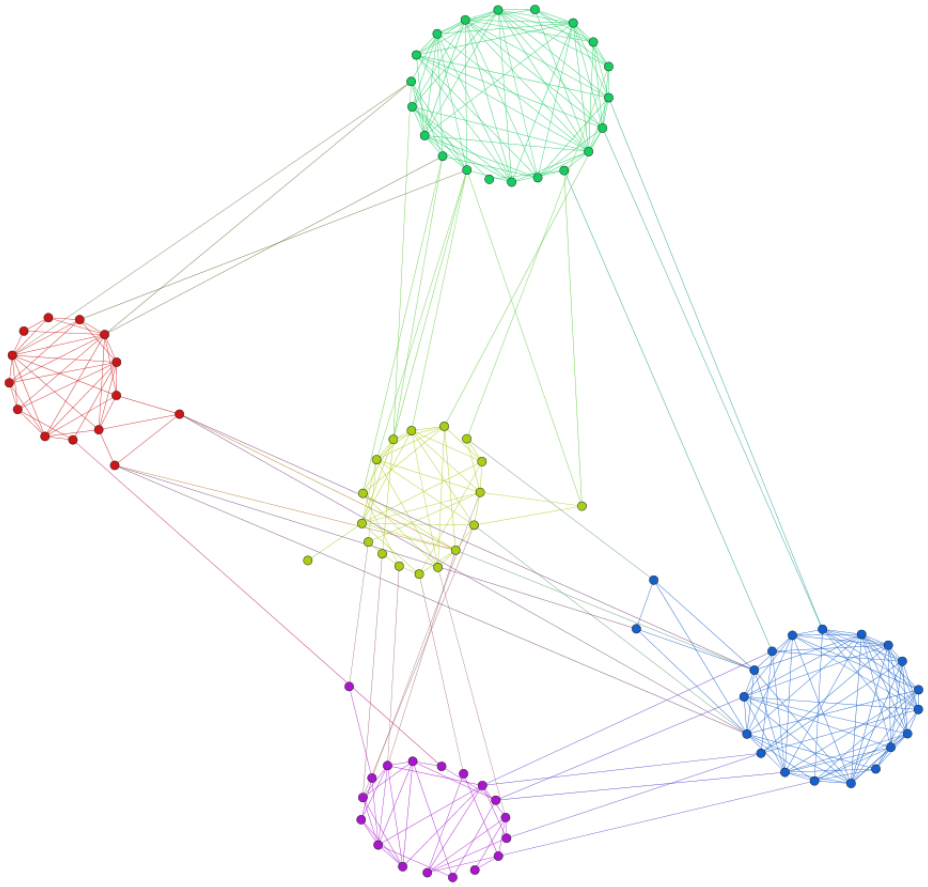

Figure 27: Resting-state functional brain network with five communities (90 nodes and 337 edges).

7.2.5 Co-purchased books on American politics

The network representing the books on American politics that are purchased by the same customer reveals interesting information towards people's reading habits. Nodes of the network correspond to the books in the New York Times Bestseller list in the year 2005 and two books are linked if they were purchased by the same person as compiled by Krebs [90]. The network exhibits two communities and interestingly they correspond to the interests shown by the people towards a political party. Identifying a community for a given node in this network not only helps facilitate recommendations, but also helps understand how book readers can influence a society in political campaigning. Our algorithm identified the two communities shown in Figure 28 and some nodes did not belong to both the communities.

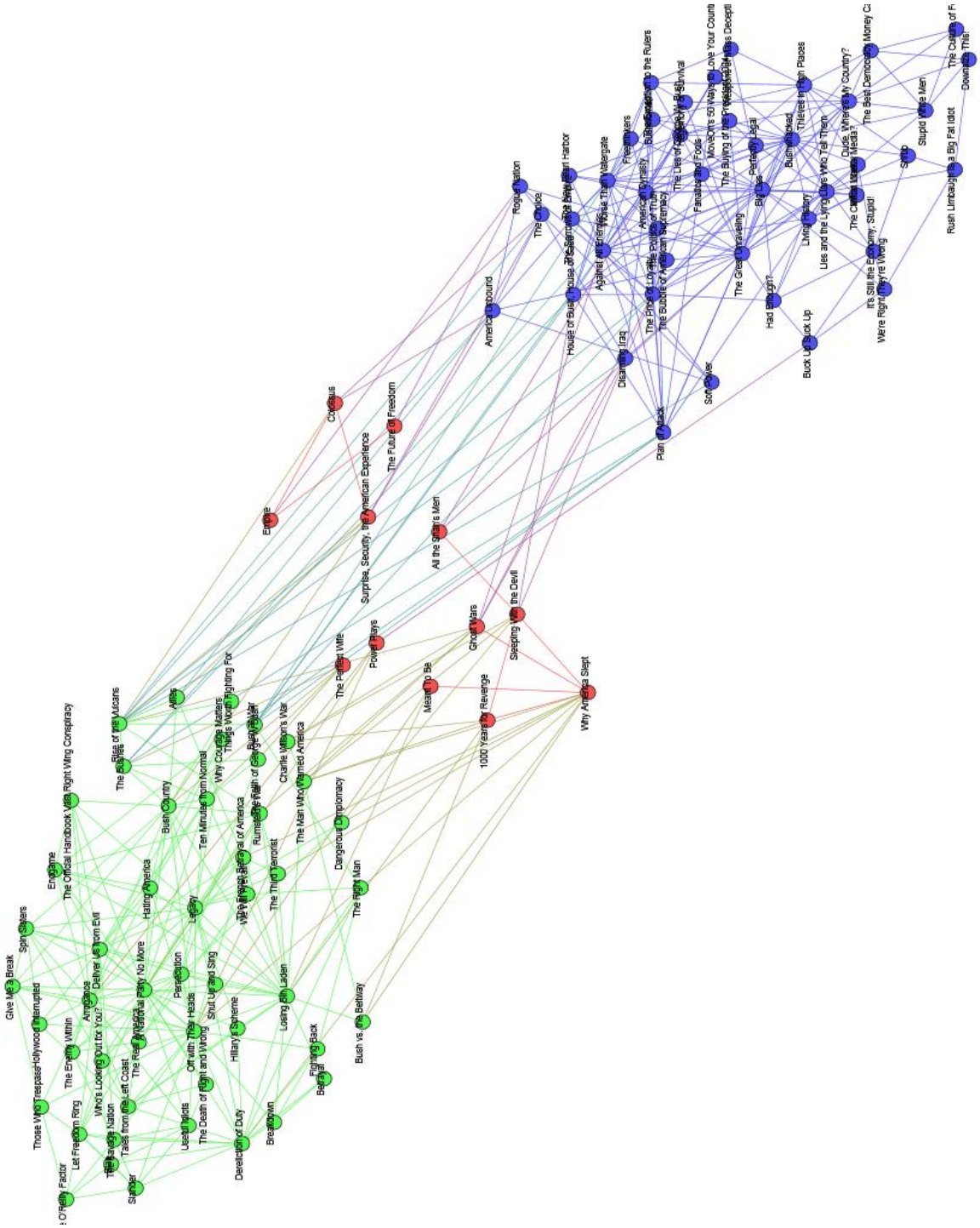

Figure 28: Co-purchased books on American politics (n = 105 and m = 441)

7.2.6 Comparing correctness of classified nodes

The results obtained from the community identification algorithms on some real-world networks were compared against the expected set of nodes (as mentioned in the literature). Different seed nodes were chosen for each of the networks randomly (all from the largest community in that network) and the union of the results obtained from each seed was used in the comparison. The Jaccard's index values of the comparison are shown in Figure 29.

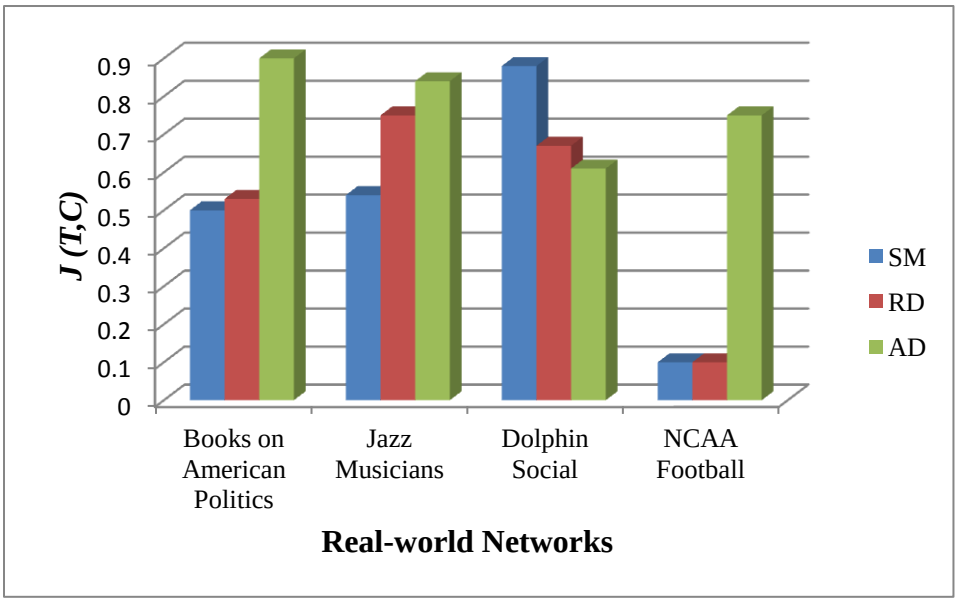

Figure 29: Jaccard index values comparing expected to obtained results in real-world networks.

Except for the dolphin's network, our algorithm identified the expected community more accurately than the other algorithms. This aberration in the result can be attributed to the presence of several nodes with degree one (leaves) in the largest community of the dolphin network (please refer to Figure 24) and our algorithm fetched only the core set of nodes with the

maximum average degree. This property becomes essential when analyzing large networks, since we are interested in the most significant members forming the community.

7.3 Biological networks

The direct correlation between the structure and function justifies the importance of efficiently identifying, detecting and extracting communities in case of biological networks. Especially in large biological networks such as the genome and neural network, identifying the best community that a given node belongs to (if there exists one) is more appropriate. Efficient algorithms to identify communities help uncover functionally significant components in a given biological network. Examples of such instances include the set of diseases a gene corresponds to in a disease network, the group of neurons associated with a particular functionality in the human brain, and the set of highly interacting proteins. The data presented (number of nodes and edges) may differ from the original datasets provided by the authors because we have eliminated self-loops and duplicate edges for analysis purposes.

Table 7: Properties of biological networks

Organism	**Network Type**	***n***	***m***	$\bar{d}$	***CC***	**Q**	**Ref.**
C. *elegans*	Neural	297	2148	14.46	0.292	0.372	[165]
C. *elegans*	Metabolic	453	2025	8.94	0.646	0.409	[54]
Human	Disease-Gene	516	1188	4.6	0.635	0.82	[74]
Human	Gene	903	6760	14.97	0.853	0.842	[74]
Yeast	Protein	2284	6646	5.81	0.13	0.567	[30]

The high modularity and clustering coefficient values reassures the existence of communities in such biological networks. Table 7 summarizes the network properties of some of the biological complex networks analyzed in the literature. The details include number of nodes (*n*),

number of edges (m), average degree ($\bar{d}$), clustering coefficient (CC), and modularity (Q). The description of these networks along with their degree distribution and the community corresponding to the node with the highest degree are as follows:

7.3.1 Caenorhabditis elegans

The complex biological processes taking place in an organism can be studied in depth from the molecular properties of their gene and protein. However, obtaining the genome sequence to perform a thorough study of the mechanisms is still in its infancy. The completely sequenced and well-annotated genome of *Caenorhabditis elegans* (*C. elegans*) marked the beginning of analyzing such metabolic networks [54, 113]. Nodes correspond to the proteins that form the molecules and their interaction is represented by an edge. The community corresponding to the node with the highest degree is shown in Figure 31.

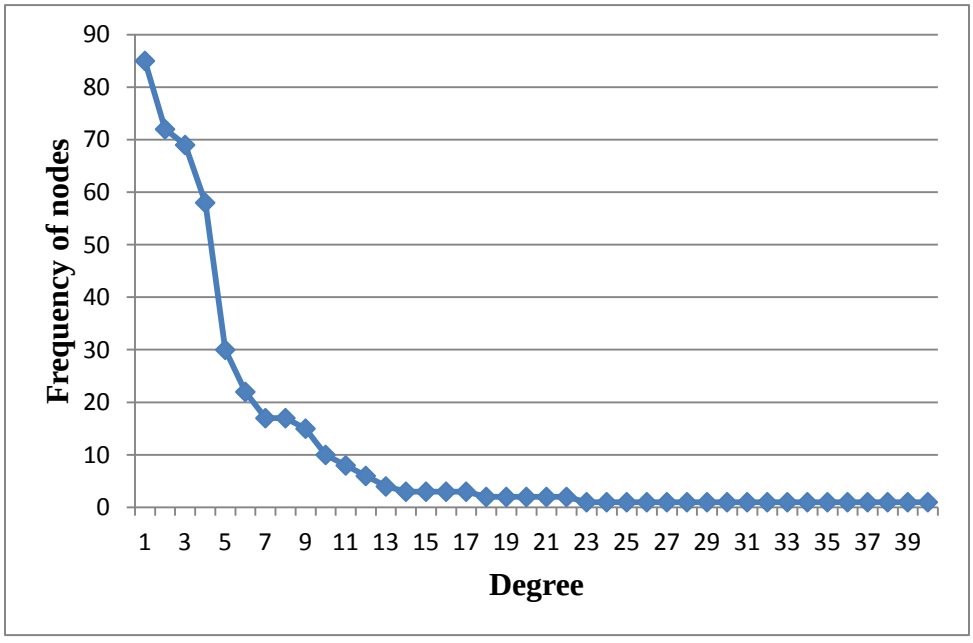

Figure 30: Power-law degree distribution of C. elegans metabolic network

Figure 31: Community identified in C. elegans metabolic network

Similarly, the neural network of C. elegans represented as a directed, weighted network was studied by White *et al*. [167]. But their network properties were analyzed by Watts and Strogatz [165], and they classified the neural network to be one of the small-world networks. We have extracted an unweighted and undirected version of the same network and identified the community corresponding to the node with the highest degree. The power-law degree

distribution and the community identified are shown in Figure 32 and Figure 33 respectively. It is to be noted that C. elegans is the only organism to have its neural network completely mapped.

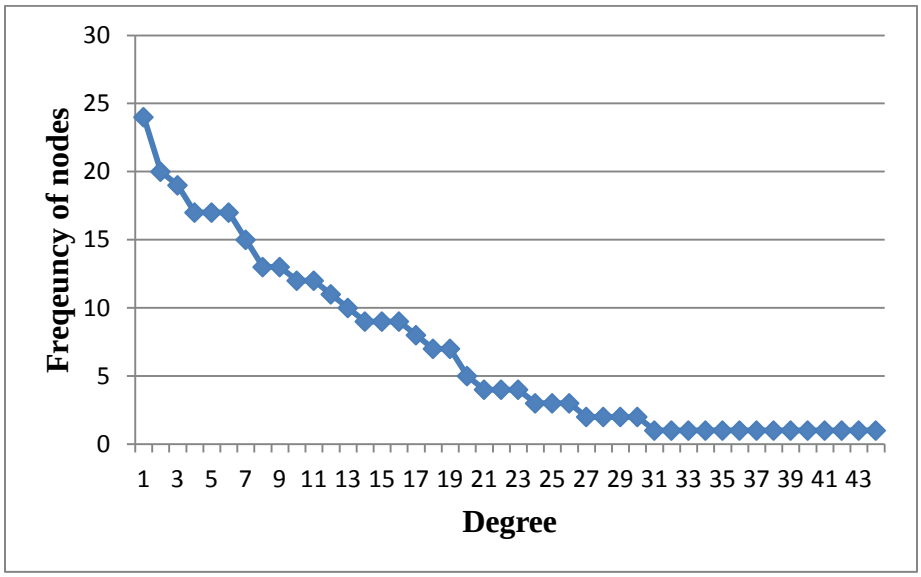

Figure 32: Power-law degree distribution of C. elegans neural network

7.3.2 Human disease network

Modeling disorders and their respective disease genes as network helps unravel the common origins of gene associations in human beings. There are two different ways of observing the disease networks obtained from human disorders [74]. In the "human disease network" nodes represent the disorders and two nodes are connected if they have at least one gene in which the mutations are associated with both disorders. In the "disease gene network", the disease genes correspond to the nodes and two genes are linked if they are associated with the same disorder. Though other network properties have been mentioned in the literature, the communities obtained from the nodes with highest degree are depicted in the Figure 34 and

Figure 35 below. The set of diseases caused by a particular gene and set of relative disorders occurring as a result of a particular disease can be easily identified by applying the average degree based technique onto these networks.

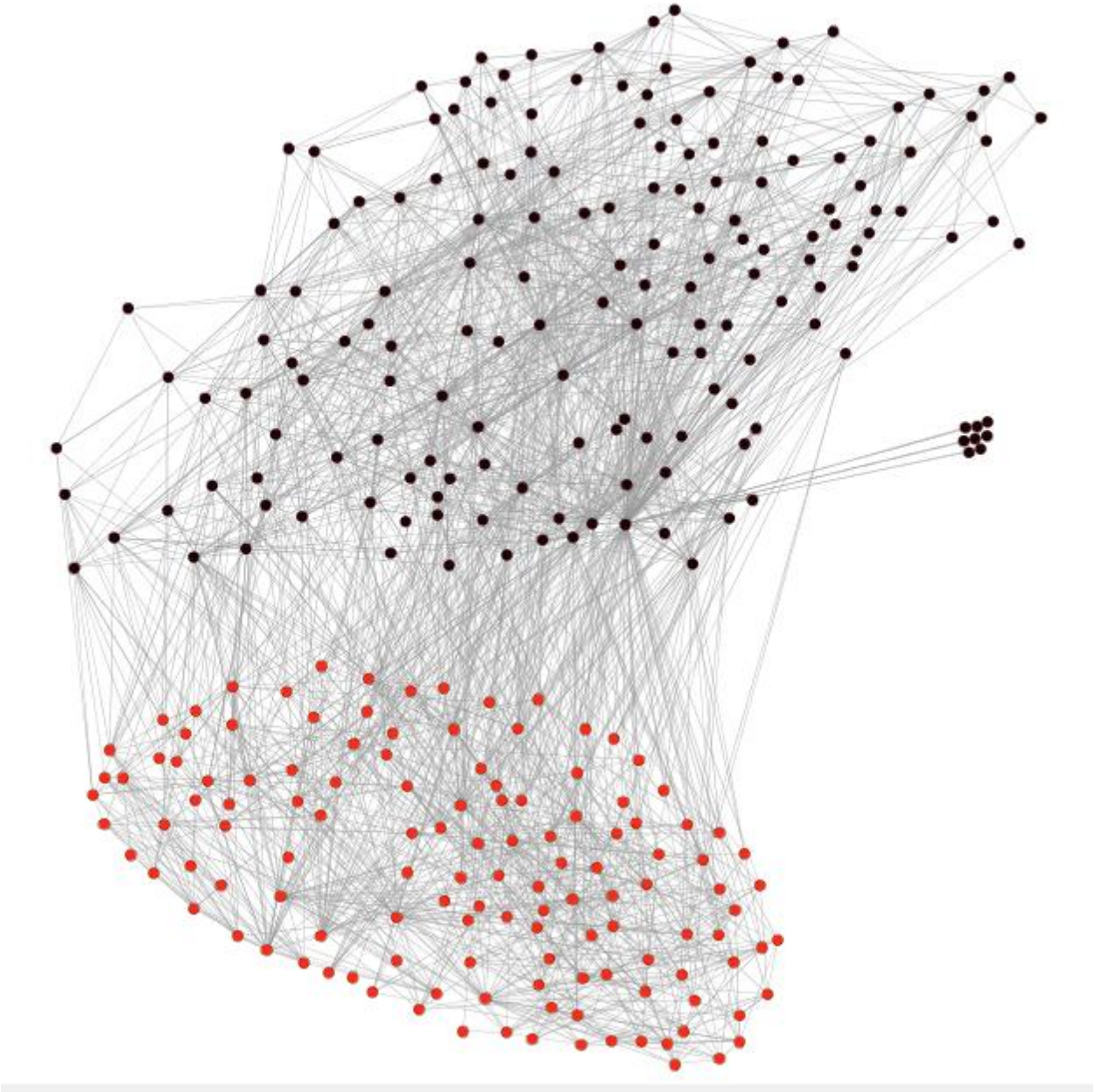

Figure 33: Community identified in C. elegans neural network

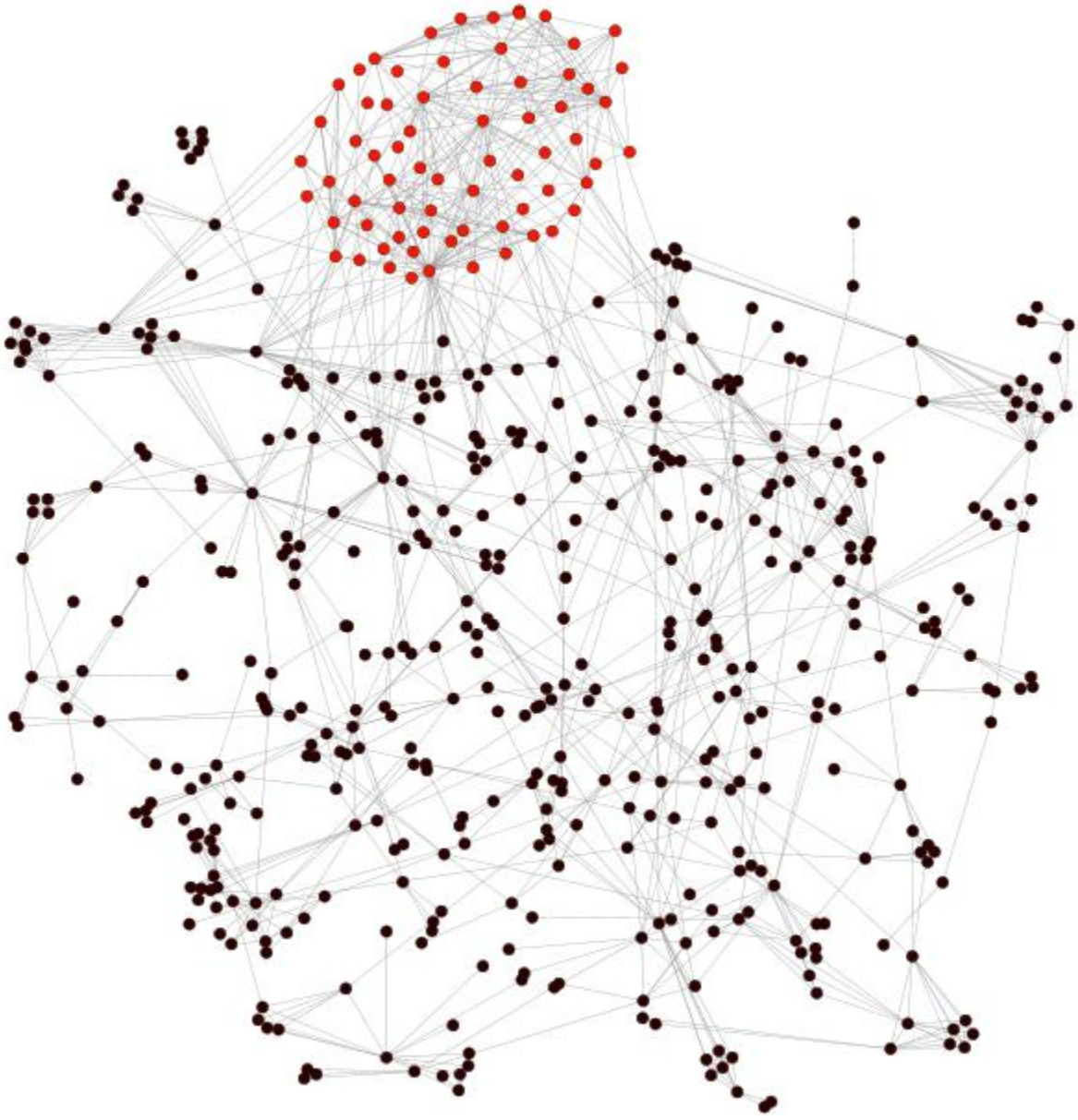

Figure 34: Community identified in human disease network

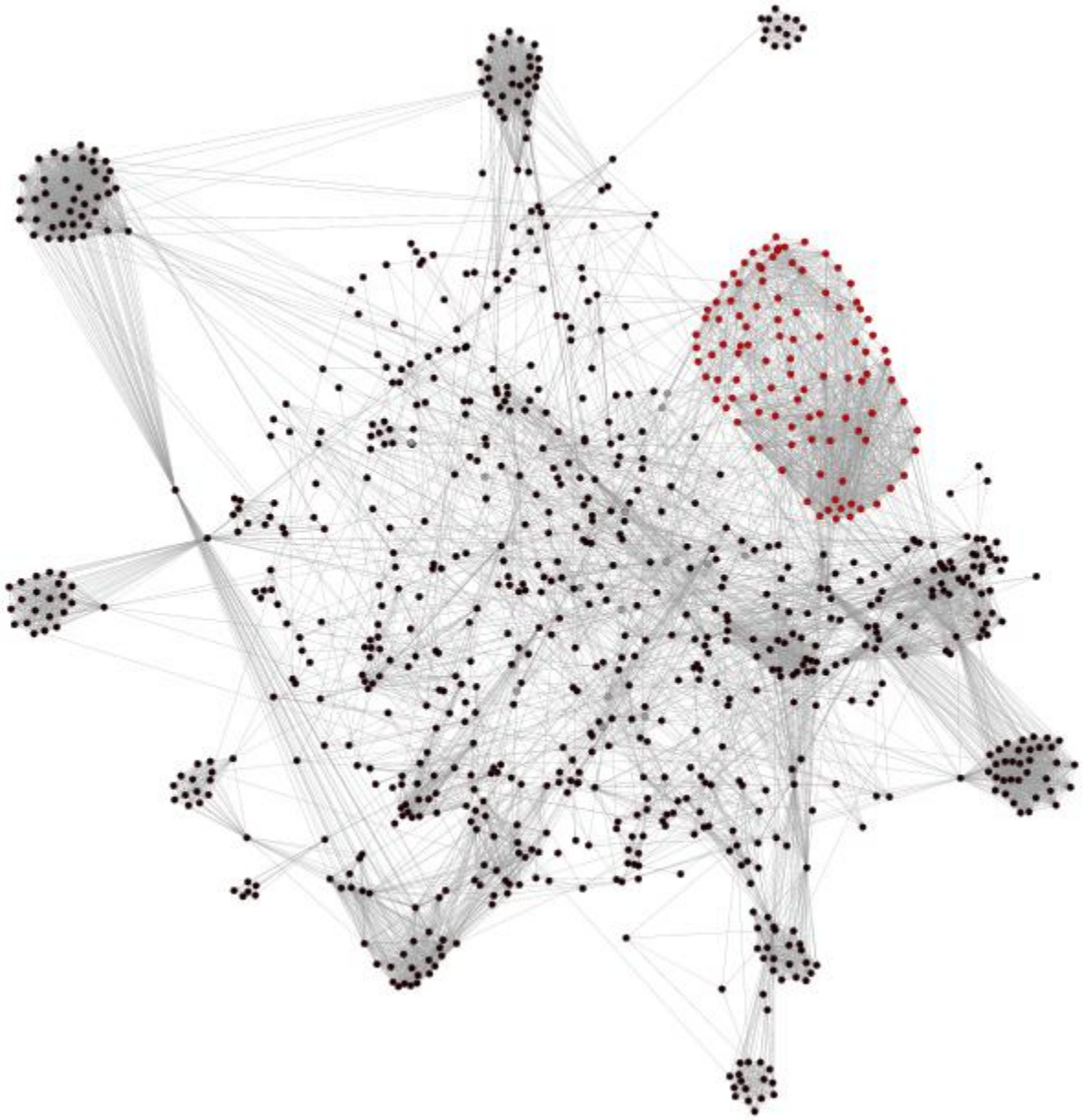

Figure 35: Community identified in disease gene network

7.3.3 Saccharomyces cerevisiae

The network properties of the Protein-Protein interaction network of *Saccharomyces cerevisiae* (Yeast) has been studied in detail by several authors [30, 84, 124]. The network is not as sparse as the other biological networks and hence the community identified contains almost the entire connected component as shown in Figure 36.

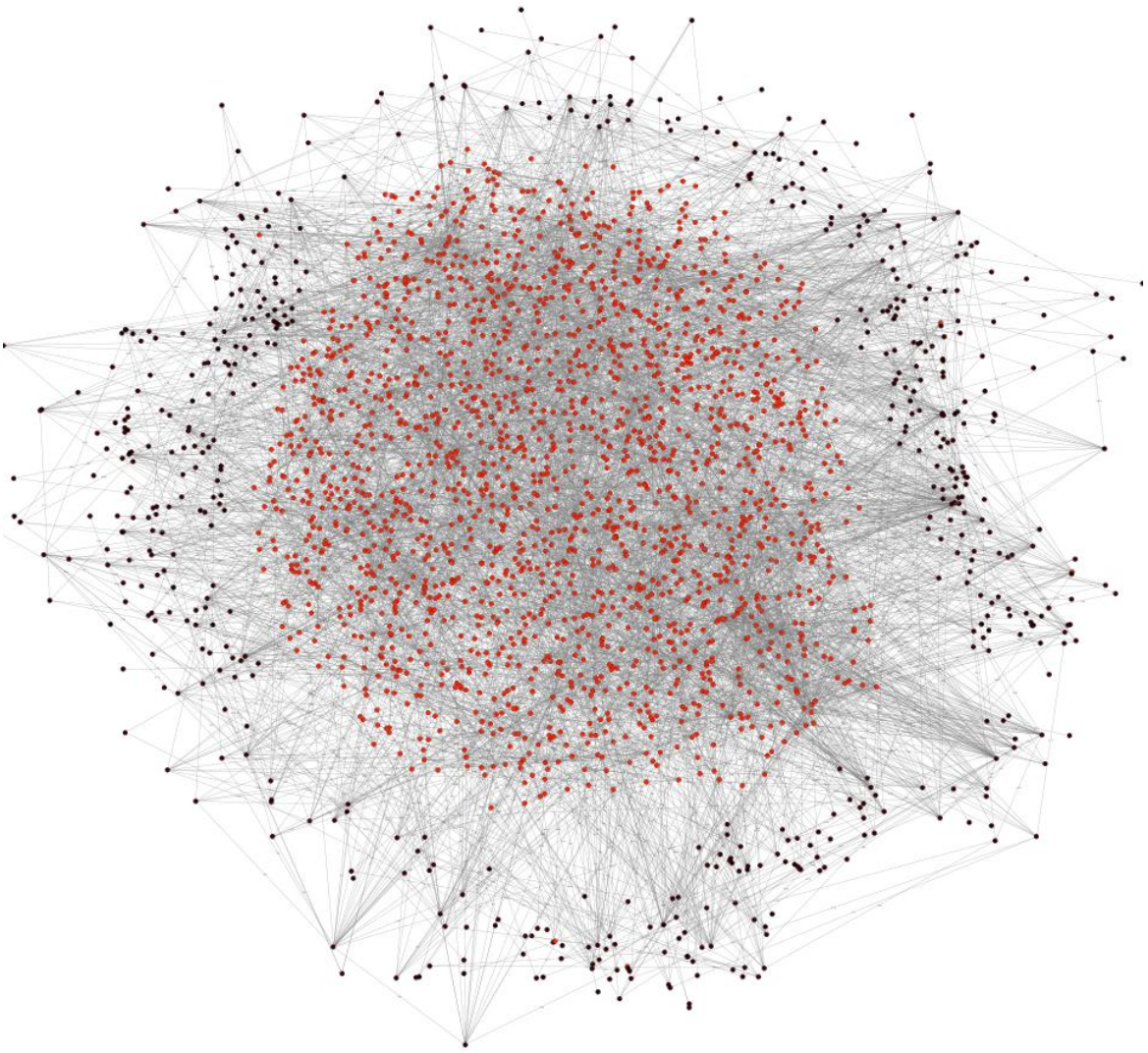

Figure 36: Community identified in Protein interaction network of Yeast

CHAPTER 8: CONCLUSION AND FUTURE DIRECTIONS

Discovering and identifying communities in large real-world networks are essential in understanding the associative behavior of the nodes. Despite the several algorithms proposed to discover communities in complex networks, the real-world networks almost always do not require extraction of all the communities. The primary objective of the current research is to devise fast and efficient methods to identify communities. This section provides suggestions for future directions in community identification and a summary of the current research.

8.1 Parallel algorithms

The novel average degree based algorithm being faster and accurate than the existing techniques would definitely serve better in case of large complex networks. However, the enormity of the complex networks necessitates the possibility of exploring the usage of multiple processors to work simultaneously and speed up the process. The key factors in designing such a parallel algorithm for community identification are data distribution and concurrent execution. Initial breadth-first exploration of the graph can be done in parallel [70, 170], but the subsequent removal of the nodes with minimal degree at each stage would cause issues.

In case of shared memory architecture, the processors can choose the minimum degree value from its block of the data and place it in a shared variable (by comparing with a value already placed by another processor). Then each processor can remove the nodes with this minimum degree value in their portion of the data. After the removal, the degree of the remaining nodes needs to be updated synchronously. However, in case of message passing architecture (distributed memory) all the processors will find out the local minimum degree

value per iteration and communicate among others to vote for the global minimum. Then each processor can remove the nodes with that minimum value locally. Computing the average degree of the induced subgraph during each stage of the algorithm would also prove expensive in terms of communication time.

However, community discovery in parallel is quite straight forward. Each process can take different input seed node and identify the community it belongs to. This would lead to discovering communities in parallel, provided the seed nodes are chosen from different communities.

8.2 Quality of a community

In this research, we have defined a community in terms of average degree and relative density that takes into account the relative dense nature of these structures in sparse neighborhood. However, an accepted quantitative definition of a community is still under debate. This makes it difficult to evaluate the algorithmic results and presence of communities in complex networks. The modularity metric [113, 117] discussed earlier has been used as a standard to compare the quality of communities in the literature. But computing the modularity of a community requires the knowledge of all the communities in the given graph. This is not feasible in case of complex networks.

8.3 Applications

Applications in the real-world scenario that requires identifying membership (community) for a given node is aplenty. Examples of such scenarios include identifying

synonyms for a given word in a linguistic network [147], finding the set of pages a given webpage has hyperlinks to (in a web graph) [23], identifying the neurons with similar functionality in the human connectome [151], and spotting the related genes corresponding to a given disease [74]. Here we mention two applications that can take advantage of community identification algorithms to a great extent.

8.3.1 *Recommendation networks*

A popular real-world application that can take advantage of network modeling is the music recommendation service [35, 36, 97]. Several online music recommendation engines such as Pandora, Musicovery, and iLike have gained popularity recently because of their ability to create playlists based on user selection. Based on the attributes of the songs such as genre, artist, acoustics, musician, etc., an edge between two songs in the music database can include weights. Community identification algorithms discussed in this work for unweighted graphs can be expanded to such weighted graphs by extracting induced subgraphs with maximum weighted average degree.

A similar network that can take advantage of community identification based recommendation is the co-purchasing network [98, 149]. Each node of the network corresponds to an item (typically listed and purchased from an online catalog) and an edge exists between two items if both items were bought by the same customer. This network resembles the co-purchased books on American politics discussed earlier. But several other factors such as the category of the item and price range can be used as weights for the edges. A community identified in such a network would correspond to the set of items the customer is most likely to buy next.

8.3.2 *Content delivery networks*

Content Delivery Networks (CDNs) or Content Distribution Networks replicate web content and deploy content servers in multiple locations often over multiple backbones and ISPs. More formally, CDN is a collaborative collection of network elements spanning the internet, where content is replicated over several mirrored web servers in order to perform transparent and effective delivery of content to the end users [122, 131]. Content providers contract with CDNs to host and distribute their content. Akamai and Limelight are two of the best CDNs distributing contents of well known content providers such as YouTube [Limelight], Microsoft, Yahoo [Akamai] etc. Fetching and replicating content onto the nearest ISP requires not only the knowledge of hyperlinks the user has visited but also the underlying connectivity among different web pages. Identifying community corresponding to the web page visited can help fetch these related web pages efficiently, thus improving user experience in terms of speed.

8.4 Ranking

Identifying a set of related nodes for a given seed has significant applications as elaborated above. But prioritizing the obtained neighbors based on their connectivity proves pivotal especially in applications such as music and movie recommendations. Ranking and categorizing the identified nodes can enhance the search techniques and provide efficient results in online search engines. Ranking in clusters has been researched in data mining to a great extent. But they look for similarity among nodes. Ranking based on the connections among the nodes would possess greater value in real-world applications.

8.5 Summary

In this dissertation the properties of complex networks and ways of mining cohesive dense structures in such networks has been investigated. A comprehensive taxonomy of the techniques to discover and detect communities, with emphasis on key algorithms has been discussed in detail. An in-depth survey of the existing community identification algorithms has also been presented. A novel greedy heuristic based on relative density of the induced subgraph, which can be used to identify communities in real-world networks efficiently is presented. We have also listed and analyzed the various community definitions in the literature. An improved definition of a community based on the average degree of the subgraph is discussed and a faster linear time divisive algorithm for identifying communities in large complex networks is presented. The performance of the algorithms on synthetic benchmark graphs and several real-world networks is also examined and comparative results are presented. Finally some related problems for future research in this topic have been stated.

APPENDIX
LIST OF SYMBOLS

a_i	fraction of edges incident on all vertices in community *i*
A	Adjacency matrix of the graph
B_C	Boundary nodes of community C
B_{ij}	Boundary adjacency matrix value at row *i* and column *j*
$C(V_C, E_C)$	Community
$CC(G)$	Clustering coefficient of graph *G*
$CC(u)$	Clustering coefficient of node *u*
C_i^I	Information centrality value for an edge incident on node *i*
$C_{i,j}^g$	edge-clustering coefficient for edge (*i*, *j*).
D	Diagonal matrix of the graph
d_i	Degree of node *i*
d_{ij}	shortest path length between nodes *i* and *j*
$\bar{d}_{V'}$	Average degree of the vertex set V'
d_{ext}	Number of external edges from the Community
d_i^{in}	Internal degree of node *i*
d_{int}	Number of internal edges within a Community
d_i^{out}	External degree of node *i*
E	Edges of *G*
e_{ij}	*k* X *k* symmetric matrix entry - fraction of the edges that link the vertices in community *i* to vertices in community *j*
f	Product of external density and relative density
$G\ (V, E)$	Unweighted, Undirected Graph
H	Hamiltonian of the graph
D_i	Total emerging degree at level *i*
K_n	Complete graph of *n* vertices
$k_{r(i)}$	contribution of node *i*'s degree on placing it to a partition *r*
L	Laplacian of the graph
m	Number of edges

n	Number of vertices
N_C	Nuclei nodes of Community C
$N\,[U]$	Union of closed neighborhood of each $u \in U$
$N[u]$	Closed Neighborhood of vertex u
$P(r)$	Proportion of nodes with degree r in a graph
Q	Modularity
Q_l	Local modularity
q_{i}	modularity value corresponding to node i
S	Subset of vertices
$S_{i,j}^{g}$	number of possible cyclic structures of order g
$T(G)$	Transitivity of graph G
V	Vertices of G
w_{ij}	influence of neighbor j in updating i's angle
$Z_{i,j}^{(g)}$	number of cycles of order g on the edge (i, j)
α	Input parameter that decides the stopping criterion in l-shell spreading algorithm
γ	Power-law coefficient
$\Gamma(k,u)$	Set of vertices within a neighborhood k from vertex u
δ	Minimum degree of the graph
$\eta_i(t)$	moving rate of node i at time step t
$\Theta_i(t)$	angle value of node i at time step t
λ_{i}	relative contribution of node i to the community structure (= $q_{\mathrm{i}}/d_{\mathrm{i}}$)
σ_i	Potts spin value $1 \leq \sigma_i \leq q$
$\delta(G)$	Density of graph G
$\partial(g)$	Nodes adjacent to the induced subgraph g but not within g
δ_E	External density of a subgraph
δ_I	Internal density of a subgraph
δ_R	Relative density of a subgraph

LIST OF REFERENCES

[1] "Division I FBS Schedules (http://www.fbschedules.com/ncaa/ncaa-football-schedules.php)."

[2] "ESPN NCAA football Schedule (http://espn.go.com/college-football/schedule)."

[3] N. R. C. (U.S.), *Network science*: National Research Council (U.S.), National Academies Press, 2005.

[4] L. A. Adamic, R. M. Lukose, A. R. Puniyani *et al.*, "Search in power-law networks," *Phys. Rev. E*, vol. 64, no. 046135, 2001.

[5] R. D. Alba, "A graph-theoretic definition of a sociometric clique," *The Journal of Mathematical Sociology*, vol. 3, no. 1, pp. 113 - 126, 1973.

[6] R. Albert, and A. Barabási, "Topology of evolving networks: Local events and universality," *Phys. Rev. Lett.*, vol. 85, no. 24, pp. 5234–5237 2000.

[7] R. Albert, H. Jeong, and A. Barabasi, "Diameter of the World-Wide Web," *Nature*, vol. 401, no. 6749, pp. 130-131, 1999.

[8] E. Almaas, R. V. Kulkarni, and D. Stroud, "Characterizing the structure of small-world networks," *Phys. Rev. Lett.*, vol. 88, no. 9, pp. 098101, 2002.

[9] L. A. N. Amaral, and M. Barthélemy, "Complex systems: Challenges, successes and tools," *Advances in condensed matter and statistical physics*, vol. 1, pp. 3-36: Nova Publishers, 2004.

[10] L. A. N. Amaral, A. Scala, M. Barthélemy *et al.*, "Classes of small-world networks," *Proc. of Natl. Acad. Sci. USA*, vol. 97, no. 21, pp. 11149-11152, October 10, 2000.

[11] A. Arenas, A. Fernández, and S. Gómez, "Analysis of the structure of complex networks at different resolution levels," *New Journal of Physics*, vol. 10, no. 5, pp. 053039, 2008.

[12] J. P. Bagrow, and E. M. Bollt, "Local method for detecting communities," *Phys. Rev. E*, vol. 72, no. 4, pp. 046108, 2005.

[13] H. Balakrishnan, and N. Deo, "Discovering communities in complex networks," in Proc. of 44th ACMSE. pp. 280 - 285, 2006.

[14] A. Barabási, *Linked: The new science of networks*: Basic Books, Perseus, Cambridge, MA, 2002.

[15] A. Barabási, “Scale-free networks: A decade and beyond,” *Science,* vol. 325, no. 5939, pp. 412 - 413, July 2009.

[16] E. R. Barnes, “An algorithm for partitioning the nodes of a graph,” *SIAM J. Alg. Discr. Meth.*, vol. 3, pp. 541, 1982.

[17] A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical processes on complex networks*, Cambridge, UK: Cambridge University Press, 2008.

[18] A. Barrat, and M. Weigt, “On the properties of small-world network models,” *Eur. Phys. J. B,* vol. 13, no. 3, pp. 547-560, 2000.

[19] M. Barthélémy, and L. A. N. Amaral, “Small-world networks: Evidence for a crossover picture,” *Phys. Rev. Lett.,* vol. 82, no. 15, pp. 3180, 1999.

[20] Bastian M., Heymann S., and J. M., "Gephi: an open source software for exploring and manipulating networks," in International AAAI Conference on Weblogs and Social Media., 2009.

[21] V. Batagelj, and A. Mrvar, "Pajek — Analysis and Visualization of Large Networks: Graph Drawing Software," Mathematics and Visualization, vol. 1, pp. 77-103: Springer Berlin Heidelberg, 2004.

[22] P. S. Bearman, J. Moody, and K. Stovel, “Chains of affection: The structure of adolescent romantic and sexual networks,” *American J. of Sociology,* vol. 110, no. 1, pp. 44-91, 2004.

[23] V. Blondel, J. Guillaume, R. Lambiotte *et al.*, “Fast unfolding of communities in large networks,” *J. Stat. Mech.*, vol. 10, no. P10008, 2008.

[24] S. Boccaletti, V. Latora, Y. Moreno *et al.*, “Complex networks: Structure and dynamics,” *Physics Reports,* vol. 424, no. 4-5, pp. 175-308, 2006.

[25] B. Bollobás, “The diameter of random graphs ” *Trans. Amer. Math. Soc. ,* vol. 267, pp. 41-52, 1981.

[26] B. Bollobás, *Modern Graph Theory*: Springer, 1998.

[27] R. C. Brigham, R. D. Dutton, T. W. Haynes *et al.*, “Powerful alliances in graphs,” *Discrete Mathematics,* vol. 309, no. 8, pp. 2140-2147, 2009.

[28] A. Broder, R. Kumar, F. Maghoul *et al.*, “Graph structure in the Web,” *Computer Networks*, vol. 33, no. 1-6, pp. 309-320, 2000.

[29] A. Z. Broder, S. C. Glassman, M. S. Manasse *et al.*, “Syntactic clustering of the Web,” *Computer Networks and ISDN Systems,* vol. 29, no. 8-13, pp. 1157-1166, 1997.

[30] D. Bu, Y. Zhao, L. Cai *et al.*, "Topological structure analysis of the protein–protein interaction network in budding yeast," *Nucleic Acids Research,* vol. 31, no. 9, pp. 2443-2450, May 1, 2003.

[31] M. Buchanan, *Nexus: Small worlds and the groundbreaking science of networks*: W. W. Norton & Company, 2002.

[32] G. Caldarelli, and A. Vespignani, *Large scale structure and dynamics of complex networks: from information technology to finance and natural science,* Singapore: World Scientific, 2007.

[33] A. Cami, H. Balakrishnan, N. Deo *et al.*, "On the complexity of finding optimal global alliances," *J. of Combinatorial Math. and Combinatorial Comp.,* vol. 58, pp. 23-32, 2006.

[34] A. Cami, and N. Deo, "Techniques for analyzing dynamic random graph models of web-like networks: An overview," *Networks,* vol. 51, no. 4, pp. 211 - 255, 2007.

[35] P. Cano, O. Celma, M. Koppenberger *et al.*, "The Topology of Music Recommendation Networks," *arXiv:physics/0512266v1,* 2005.

[36] P. Cano, M. Koppenberger, and N. Wack, "Content-based music audio recommendation," in Proceedings of the 13th annual ACM international conference on Multimedia, Hilton, Singapore, 2005, pp. 211-212.

[37] S. Chakrabarti, A. Frieze, and J. Vera, "The influence of search engines on preferential attachment," in Proc. of 16th annual ACM-SIAM symposium on Discrete algorithms, Vancouver, British Columbia, 2005.

[38] D. Chen, Y. Fu, and M. Shang, "A fast and efficient heuristic algorithm for detecting community structures in complex networks," *Physica A: Statistical Mechanics and its Applications,* vol. 388, no. 13, pp. 2741-2749, July 2009.

[39] J. Chen, and B. Yuan, "Detecting functional modules in the yeast protein-protein interaction network," *Bioinformatics,* vol. 22, no. 18, pp. 2283-2290, September 15, 2006.

[40] J. Chen, O. Zaiane, and R. Goebel, "Local community identification in social networks," in Intl. Conf. on Adv. in Social Networks Analysis and Mining (ASONAM), . pp. 237-242, 2009.

[41] F. Chung, and H. Lu, *Complex graphs and networks*: AMS Bookstore, 2006.

[42] F. Chung, L. Lu, T. G. Dewey *et al.*, "Duplication models for biological networks," *Journal of Computational Biology,* vol. 10, no. 5, pp. 677-687, 2003.

[43] A. Clauset, “Finding local community structures in networks,” *Phys. Rev. E,* vol. 72, no. 026132, 2005.

[44] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E,* vol. 70, pp. 066111, 2004.

[45] L. Costa, "Hub-based community finding," *arXiv:cond-mat/0405022v1*, 2004.

[46] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc, 1974.

[47] N. Deo, and A. Cami, “Preferential deletion in dynamic models of web-like networks,” *Info. Processing Lett.,* vol. 102, no. 4, pp. 156-162, 2007.

[48] R. Diestel, *Graph theory*: Birkhäuser, 2006.

[49] S. N. Dorogovtsev, and J. F. F. Mendes, *Evolution of networks: From biological nets to the Internet and WWW*: Oxford University Press, 2003.

[50] S. N. Dorogovtsev, and J. F. F. Mendes, “Scaling properties of scale-free evolving networks: Continuous approach,” *Phys. Rev. E,* vol. 63, no. 5, pp. 056125, 2001.

[51] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, “Structure of growing networks with preferential linking,” *Phys. Rev. Lett.,* vol. 85, no. 21, pp. 4633, 2000.

[52] Y. Dourisboure, F. Geraci, and M. Pellegrini, “Extraction and classification of dense communities in the web,” in Proc. of the 16th Intl. conf. on World Wide Web, Banff, Alberta, Canada, 2007, pp. 461 - 470.

[53] N. Du, B. Wu, and B. Wang, “Community detection in complex networks,” *J. Comp. Sci. & Tech. ,* vol. 21, pp. 672 - 683, 2006.

[54] J. Duch, and A. Arenas, “Community detection in complex networks using external optimization,” *Phys. Rev. E,* vol. 72, no. 027104, 2005.

[55] R. Durrett, *Random graph dynamics*: Cambridge University Press, 2007.

[56] H. Ebel, L.-I. Mielsch, and S. Bornholdt, “Scale-free topology of e-mail networks,” *Phys. Rev. E,* vol. 66, no. 3, pp. 035103, 2002.

[57] V. M. Eguíluz, and K. Klemm, “Epidemic threshold in structured scale-free networks,” *Phys. Rev. Lett.,* vol. 89, no. 10, pp. 108701, 2002.

[58] P. Erdös, and A. Rényi, “On random graphs,” *Publ. Math. Debrecen,* vol. 6, pp. 290-297, 1959.

[59] T. S. Evans, “Clique graphs and overlapping communities,” *J. Stat. Mech.* , vol. 2010, no. P12037, 2010.

[60] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the Internet topology,” in Proc. of conf. on Applications, technologies, architectures, and protocols for computer communication, Cambridge, Massachusetts, United States, 1999, pp. 251 - 262

[61] G. W. Flake, S. Lawrence, and C. L. Giles, “Efficient identification of Web communities,” in Proc. of 6th ACM SIGKDD Intl. conf. on Knowledge discovery and data mining, Boston, Massachusetts, United States, 2000, pp. 150 - 160

[62] G. W. Flake, S. Lawrence, C. Lee Giles *et al.*, “Self-organization and identification of Web communities,” *Computer,* vol. 35, no. 3, pp. 66-71, 2002.

[63] A. D. Flaxman, A. M. Frieze, and J. Vera, “A geometric preferential attachment model of networks,” *Internet Math,* vol. 3, no. 2, pp. 187-205, 2006.

[64] S. Fortunato, “Community detection in graphs,” *Physics Report,* vol. 486, no. 3 - 5, pp. 75 - 174, 2010.

[65] S. Fortunato, and M. Barthélemy, “Resolution limit in community detection,” *Proc. of Natl. Acad. Sci. USA,* vol. 104, no. 1, pp. 36-41, January 2, 2007.

[66] S. Fortunato, V. Latora, and M. Marchiori, “Method to find community structure based on Information centrality,” *Phys. Rev. E,* vol. 70, no. 056104, 2004.

[67] L. C. Freeman, *The development of social network analysis: a study in the sociology of science*: Empirical Press, 2004.

[68] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry,* vol. 40, no. 1, pp. 35-41, 1977.

[69] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A guide to the theory of NP-Completeness*: Freeman, New York, USA, 1979.

[70] R. K. Ghosh, and G. P. Bhattacharjee, “Parallel breadth-first search algorithms for trees and graphs,” *International Journal of Computer Mathematics,* vol. 15, no. 1-4, pp. 255-268, 1984.

[71] E. N. Gilbert, “Random Graphs,” *The Annals of Mathematical Statistics,* vol. 30, no. 4, pp. 1141-1144, 1959.

[72] M. Girvan, and M. E. J. Newman, “Community structure in social and biological networks,” *Proc. of Natl. Acad. Sci. USA,* vol. 99, no. 12, pp. 7821-7826, June, 2002.

[73] P. M. Gleiser, and L. Danon, "Community Structure in Jazz," *Advances in Complex Systems,* vol. 6, no. 4, pp. 565-573, 2003.

[74] K.-I. Goh, M. E. Cusick, D. Valle *et al.*, "The human disease network," *Proceedings of the National Academy of Sciences,* vol. 104, no. 21, pp. 8685-8690, May 22, 2007.

[75] K. I. Goh, E. Oh, H. Jeong *et al.*, "Classification of scale-free networks," *Proc. of Natl. Acad. Sci. USA,* vol. 99, no. 20, pp. 12583-12588, 2002.

[76] K. I. Goh, E. Oh, B. Kahng *et al.*, "Betweenness centrality correlation in social networks," *Phys. Rev. E,* vol. 67, no. 1, pp. 017101, 2003.

[77] S. Gregory, "Finding overlapping communities using disjoint community detection algorithms," *Studies in Computational Intelligence,* vol. 207, pp. 47 - 61, 2009.

[78] P. Hagmann, L. Cammoun, X. Gigandet *et al.*, "Mapping the Structural Core of Human Cerebral Cortex," *PLoS Biol,* vol. 6, no. 7, pp. e159, 2008.

[79] E. Hartuv, and R. Shamir, "A clustering algorithm based on graph connectivity," *Info. Processing Lett.,* vol. 76, no. 4-6, pp. 175-181, 2000.

[80] Y. He, J. Wang, L. Wang *et al.*, "Uncovering Intrinsic Modular Organization of Spontaneous Brain Activity in Humans," *PLoS ONE,* vol. 4, no. 4, pp. e5226, 2009.

[81] J. Hopcroft, O. Khan, B. Kulis *et al.*, "Tracking evolving communities in large linked networks," *Proc. of Natl. Acad. Sci. USA,* vol. 101, pp. 5249-5253, 2004.

[82] Y. Hu, H. Chen, P. Zhang *et al.*, "Comparative definition of community and corresponding identifying algorithm," *Phys. Rev. E,* vol. 78, no. 2, pp. 026121, 2008.

[83] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.,* vol. 31, no. 3, pp. 264-323, 1999.

[84] H. Jeong, S. P. Mason, A. L. Barabasi *et al.*, "Lethality and centrality in protein networks," *Nature,* vol. 411, no. 6833, pp. 41-42, 2001.

[85] F. Karinthy, "Chains," Everything is Different, 1929.

[86] B. W. Kernighan, and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Tech. J.,* vol. 49, pp. 291, 1970.

[87] J. M. Kleinberg, "The convergence of social and technological networks," *Comm. of the ACM,* vol. 51, no. 11, pp. 66-72, 2008.

[88] E. V. Koonin, Y. I. Wolf, and G. P. Karev, *Power laws, scale-free networks and genome biology*: Birkhäuser, 2006.

[89] A. E. Krause, K. A. Frank, D. M. Mason *et al.*, “Compartments revealed in food-web structure,” *Nature*, vol. 426, no. 6964, pp. 282-285, 2003.

[90] V. Krebs. "http://www.orgnet.com/index.html."

[91] M. Kretzschmar, and M. Morris, “Measures of concurrency in networks and the spread of infectious disease,” *Mathematical Biosciences*, vol. 133, no. 2, pp. 165-196, 1996.

[92] P. Kristiansen, S. M. Hedetniemi, and S. T. Hedetniemi, “Alliances in Graphs,” *J. of Combinatorial Math. and Combinatorial Comp.*, vol. 48, pp. 157-178, 2004.

[93] J. M. Kumpula, J. Saramäki, K. Kaski *et al.*, “Limited resolution in complex network community detection with Potts model approach ” *Eur. Phys. J. B*, vol. 56, no. 1, pp. 41 - 45, 2007.

[94] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Phys. Rev. E*, vol. 78, pp. 046110, 2008.

[95] A. Lancichinetti, F. Radicchi, J. J. Ramasco *et al.*, “Finding Statistically Significant Communities in Networks,” *PLoS ONE*, vol. 6, no. 4, pp. e18961, 2011.

[96] E. A. Leicht, and M. E. J. Newman, “Community structure in directed networks,” *Phys. Rev. Lett.*, vol. 100, no. 11, pp. 118703, 2008.

[97] J. Leskovec, A. Singh, and J. Kleinberg, "Patterns of Influence in a Recommendation Network," Lecture Notes in Computer Science, vol. 3918, pp. 380-389: Springer Berlin / Heidelberg, 2006.

[98] I. Leung, P. Hui, P. Lio *et al.*, “Towards real-time community detection in large networks,” *Phys. Rev. E*, vol. 79, pp. 066107, 2009.

[99] F. Liljeros, C. R. Edling, and L. A. N. Amaral, “Sexual networks: implications for the transmission of sexually transmitted infections,” *Microbes and Infection*, vol. 5, no. 2, pp. 189-196, 2003.

[100] F. Liljeros, C. R. Edling, L. A. N. Amaral *et al.*, “The web of human sexual contacts,” *Nature*, vol. 411, no. 6840, pp. 907-908, 2001.

[101] F. Luccio, and M. Sami, “On the Decomposition of Networks in Minimally Interconnected Subnetworks,” *Circuit Theory, IEEE Transactions on*, vol. 16, no. 2, pp. 184-188, 1969.

[102] R. Luce, and A. Perry, “A method of matrix analysis of group structure,” *Psychometrika*, vol. 14, no. 2, pp. 95-116, 1949.

[103] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," in Proc. of Intl.Conf. on Web Intelligence. pp. 233 - 239, 2006.

[104] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelligence & Agent Systems,* vol. 6, no. 4, pp. 387-400, 2008.

[105] F. Luo, Y. Yang, C.-F. Chen *et al.*, "Modular organization of protein interaction networks," *Bioinformatics,* vol. 23, no. 2, pp. 207-214, January 15, 2007.

[106] D. Lusseau, "The emergent properties of a dolphin social network," *Proc. of Royal Society of London. Series B: Biological Sciences,* vol. 270, no. Suppl 2, pp. S186-S188, November 7, 2003.

[107] D. Lusseau, K. Schneider, O. J. Boisseau *et al.*, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology,* vol. 54, no. 4, pp. 396-405, 2003.

[108] S. Milgram, "The small-world problem," *Psychology Today,* vol. 67, no. 1, 1967.

[109] R. J. Mokken, "Cliques, clubs and clans," *Quality & Quantity,* vol. 13, no. 2, pp. 161-173, 1979.

[110] M. E. J. Newman, "Detecting community structure in networks," *Eur. Phys. J. B,* vol. 38, pp. 321 - 330, 2004.

[111] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E,* vol. 69, pp. 066133, 2004.

[112] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E,* vol. 74, pp. 036104, 2006.

[113] M. E. J. Newman, "Modularity and community structure in networks," *Proc. of Natl. Acad. Sci. USA,* vol. 103, pp. 8577–8582, 2006.

[114] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics,* vol. 46, no. 5, pp. 323-351, Sept. 2005.

[115] M. E. J. Newman, "Properties of highly clustered networks," *Phys. Rev. E,* vol. 68, no. 2, pp. 026121, 2003.

[116] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review,* vol. 45, no. 2, pp. 167 - 256, 2003.

[117] M. E. J. Newman, and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E,* vol. 69, no. 026113, 2004.

[118] Z. Nikoloski, N. Deo, and L. Kucera, "Degree-correlation of a scale-free random graph process," in Proc. of Discrete Math. Theoritical Comp. Sci. pp. 239-244, 2005.

[119] T. B. S. d. Oliveira, and L. Zhao, "Complex Network Community Detection Based on Swarm Aggregation," in Proc. of the 2008 4th Intl. Conf. on Natural Computation, 2008, pp. 604-608.

[120] J.-P. Onnela, D. Fenn, S. Reid *et al.*, "A Taxonomy of Networks," *arXiv:1006.5731v2*, 2010.

[121] G. Palla, I. Derenyi, I. Farkas *et al.*, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 03607, pp. 815 - 818, 2005.

[122] G. Pallis, and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM - Personal information management*, vol. 49, no. 1, 2006.

[123] C. R. Palmer, P. B. Gibbons, and C. Faloutsos, "ANF: a fast and scalable tool for data mining in massive graphs," in Proc. of 8th ACM SIGKDD Intl. conf. on Knowledge discovery and data mining, Edmonton, Alberta, Canada, 2002.

[124] M. Pellegrini, D. Haynor, and J. M. Johnson, "Protein interaction networks," *Expert Rev. of Proteomics*, vol. 1, no. 2, pp. 239-249, 2004.

[125] S. L. Pimm, "The structure of food webs," *Theoretical Population Biology*, vol. 16, no. 2, pp. 144-158, 1979.

[126] P. Pollner, G. Palla, and T. Vicsek, "Preferential attachment of communities: The same principle, but a higher level," *Eur. Phy. Lett.*, vol. 73, no. 3, pp. 478, 2006.

[127] M. A. Porter, J. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, pp. 1082 - 1097, 1164 - 1166, Oct 2009.

[128] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 430-452, 1990.

[129] F. Radicchi, C. Castellano, F. Cecconi *et al.*, "Defining and identifying communities in networks," *Proc. of Natl. Acad. Sci. USA*, vol. 101, pp. 2658 - 2663, 2004.

[130] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E*, vol. 76, pp. 036106, 2007.

[131] B. Rajkumar, P. Mukkadim, and V. Athena, *Content Delivery Networks*: Springer, 2008.

[132] R. Real, and J. M. Vargas, "The Probabilistic Basis of Jaccard's Index of Similarity " *SYSTEMATIC BIOLOGY*, vol. 45, no. 3, pp. 380-384, 1996.

[133] K. P. Reddy, M. Kitsuregawa, P. Sreekanth *et al.*, "A graph based approach to extract a neighborhood customer community for collaborative filtering," *Lect. Notes in Comp. Sci.*, vol. 2544, pp. 188-200, 2002.

[134] S. Redner, "How popular is your paper? An empirical study of the citation distribution," *Eur. Phys. J. B,* vol. 4, no. 2, pp. 131-134, 1998.

[135] J. Reichardt, and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a Potts model," *Phys. Rev. Lett.,* vol. 93, pp. 218701, 2004.

[136] S. A. Rice, "The identification of blocs in small political bodies," *The American Political Science Review,* vol. 21, no. 3, pp. 619-627, 1927.

[137] A. W. Rives, and T. Galitski, "Modular organization of cellular networks," *Proc. of Natl. Acad. Sci. USA,* vol. 100, no. 3, pp. 1128-1133, February 4, 2003.

[138] J. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*, 2nd ed.: Routledge, 2009.

[139] P. Ronhovde, and Z. Nussinov, "An Improved Potts Model Applied to Community Detection," *In Practice,* vol. 63130, no. i, pp. 4, 2008.

[140] J. M. Scanlon, and N. Deo, "Network Communities Based on Maximizing Average Degree," *Congressus Numerantium,* vol. 190, pp. 183 - 192, 2008.

[141] S. E. Schaeffer, "Graph Clustering," *Computer Science Review,* vol. 1, no. 1, pp. 27-64, August 2007.

[142] S. E. Schaeffer, "Stochastic local clustering for massive graphs," *Proc. of 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, LNCS,* vol. 3518, pp. 354 - 360, 2005.

[143] N. Schwartz, R. Cohen, D. ben-Avraham *et al.*, "Percolation in directed scale-free networks," *Phys. Rev. E,* vol. 66, no. 1, pp. 015104, 2002.

[144] J. Scott, *Social network analysis: A handbook,* 2nd ed., London: SAGE Publications, 2000.

[145] S. B. Seidman, "Network structure and minimum degree," *Social Networks,* vol. 5, no. 3, pp. 269-287, 1983.

[146] S. B. Seidman, and B. L. Foster, "A graph-theoretic generalization of the clique concept," *The Journal of Mathematical Sociology,* vol. 6, no. 1, pp. 139 - 154, 1978.

[147] H. Shen, X. Cheng, K. Cai *et al.*, "Detect overlapping and hierarchical community structure in networks," *Physica A,* vol. 388, pp. 1706 - 1712, 2009.

[148] R. Solomonoff, and A. Rapoport, "Connectivity of random nets," *Bulletin of Mathematical Biophysics,* vol. 13, pp. 107-117, 1951.

[149] Y. Song, Z. Zhuang, H. Li *et al.*, "Real-time automatic tag recommendation," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, Singapore, Singapore, 2008.

[150] V. Spirin, and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proc. of Natl. Acad. Sci. USA,* vol. 100, no. 21, pp. 12123-12128, October 14, 2003.

[151] O. Sporns, "The human connectome: a complex network," *Annals of the New York Academy of Sciences,* vol. 1224, no. 1, pp. 109-125, 2011.

[152] D. Stephen, K. Ravi, S. M. Kevin *et al.*, "Self-similarity in the web," *ACM Trans. Internet Technol.,* vol. 2, no. 3, pp. 205-223, 2002.

[153] S. H. Strogatz, "Exploring complex networks," *Nature,* vol. 410, no. 6825, pp. 268-276, 2001.

[154] G. Su, A. Kuchinsky, J. H. Morris *et al.*, "GLay: community structure analysis of biological networks," *Bioinformatics,* vol. 26, no. 24, pp. 3135-3137, December 15, 2010.

[155] Y. Sun, B. Danila, K. Josić *et al.*, "Improved community structure detection using a modified fine-tuning strategy," *Eur. Phys. Lett.,* vol. 86, no. 2, pp. 28004, 2009.

[156] G. Szabó, M. Alava, and J. Kertész, "Clustering in Complex Networks," *Lect. Notes in Phys., Springer,* vol. 650, pp. 139 - 162, 2004.

[157] M. Vasudevan, H. Balakrishnan, and N. Deo, "Community discovery algorithms: An overview," *Congressus Numerantium,* vol. 196, pp. 127-142, 2009.

[158] M. Vasudevan, and N. Deo, "Community identification algorithm using relative edge density measure," *Congressus Numerantium,* vol. 204, pp. 147-160, 2010.

[159] A. Vázquez, "Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations," *Phys. Rev. E,* vol. 67, no. 5, pp. 056104, 2003.

[160] A. Vázquez, R. Pastor-Satorras, and A. Vespignani, "Large-scale topological and dynamical properties of the Internet," *Phys. Rev. E,* vol. 65, no. 6, pp. 066130, 2002.

[161] K. Wakita, and T. Tsurumi, "Finding Community Structure in Mega-scale Social Networks," *Analysis,* vol. 105, no. 2, pp. 9, 2007.

[162] J. Wang, X. Zuo, and Y. He, "Graph-based network analysis of resting-state functional MRI," *Frontiers in Systems Neuroscience,* vol. 4, June 7, 2010.

[163] X. Wang, G. Chen, and H. Lu, "A very fast algorithm for detecting community structures in complex networks," *Physica A,* vol. 384, no. 2, pp. 667 - 674, Oct. 2007.

[164] D. J. Watts, *Six degrees: The science of a connected age*: W. W. Norton & Company, 2004.

[165] D. J. Watts, and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature,* vol. 393, no. 6684, pp. 440-442, 1998.

[166] R. S. Weiss, and E. Jacobson, "A method for the analysis of the structure of complex organizations," *American Sociological Review,* vol. 20, no. 6, pp. 661-668, 1955.

[167] J. G. White, E. Southgate, J. N. Thompson *et al.*, "The structure of the nervous system of the nematode C. Elegans," *Phil. Trans. R. Soc. London,* vol. 314, pp. 1-340, 1986.

[168] W. Xiao, L. Peng, and B. Parhami, "On general laws of complex networks," *Complex Sciences,* vol. 4, pp. 118-124: Springer Berlin Heidelberg, 2009.

[169] X.-J. Xu, X. Zhang, and J. F. F. Mendes, "Growing community networks with local events," *Physica A: Statistical Mechanics and its Applications,* vol. 388, no. 7, pp. 1273-1278, 2009.

[170] A. Yoo, E. Chow, K. Henderson *et al.*, "A Scalable Distributed Parallel Breadth-First Search Algorithm on BlueGene/L," in Proceedings of the 2005 ACM/IEEE conference on Supercomputing, 2005, pp. 25.

[171] W. W. Zachary, "An information flow model for conflict and fission in Small Groups," *J. of Anthropological Research,* vol. 33, no. 4, pp. 452-473, 1977.