Overlapping Community Detection in Social Networks by Local Expansion

Eaint Mon Win

University of Computer Studies, Yangon

September, 2023

Overlapping Community Detection in Social Networks by Local Expansion

Eaint Mon Win

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

September, 2023

ACKNOWLEDGEMENTS

First of all, I would like to thank His Excellency, the Minister, the Ministry of Science and Technology for full facilities support during the Ph.D Course at the University of Computer Studies, Yangon.

I would like to express very special thanks to Dr. Me Me Khin, the Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance during the period of my study. Further, I would like to offer my special thanks to Dr. Mie Mie Thet Thwin, former Rector, the University of Computer Studies, Yangon, for her guidance and encouragement during the Ph.D. life study.

I am also very grateful to Dr. Ei Ei Hlaing, Rector, the University of Computer Studies, Taungoo, for her encouragement to concentrate on only research during my research work.

I am also very grateful to Dr. Win Pa Pa, Professor and Course coordinator of the Ph.D. 12th Batch, University of Computer Studies, Yangon, for her valuable advice, moral and emotional support in my research work.

I would like to express my deepest gratitude to my supervisor, Dr. Si Si Mar Win, Professor, the University of Computer Studies, Yangon, for her excellent guidance, caring, patience, and providing me with excellent ideas for doing research.

I would like to express my respectful gratitude to Daw Aye Aye Khine, Professor, Head of English Department for her valuable supports from the language point of view and pointed out the correct usage in my dissertation.

My sincere thanks also go to all my respectful Professors for giving me valuable lectures and knowledge during the Ph.D course work.

In addition, I would like to extend my appreciation and thanks to the board of examiners for valuable comments and recommending my thesis. I also thank my friends from Ph.D 12th batch for providing support and friendship that I needed.

Last but by no means least, I must express my very profound gratitude to my family for always believing in me, for providing me with unfailing support and continuous encouragement, for their endless love throughout my years of Ph.D study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them.

Abstract

Community structure is one of the main structural features of networks and detecting overlapped community structure is an important field in social network analysis. There are many methods for finding non-overlapping communities in this research area. The existing studies about overlapping do not sufficiently address the problems of the relationship between objects in overlapping regions and the roles of these objects during the formation and growth of communities. In recent years, local community detection algorithms which detect overlapped community structure have been developed. Local expansion methodologies that detect local community structure are techniques to find a community through the seed. Therefore, recent algorithms have emphasized on the locating seed rather than random seed selection. However, although the most existing algorithms could identify superior seed, their expansion strategies did not become effective and efficient strategies. Moreover, algorithms suffer unstable community structure because the influences of parameter for controlling community's resolution of fitness evaluation functions where used in community expansion process. In this research, therefore, the algorithm is modelled on local expansion strategy and designs the extended jaccrad similarity to find seed. In addition, this research formulates the optimized parameter evaluation formula to avoid the parameter influences. This work, firstly, identifies the seed or core node by using extended jaccard similarity and form initial community via seed. Then local community is detected by expanding the initial community with fitness function based on proposed optimized parameter evaluation and finally overlapped nodes are identified by merging detected local communities. In this dissertation, the algorithm is implemented by using small datasets from network data repository site and large networks from Stanford large network datasets collection. In addition to real networks, overlapping artificial benchmarks are also selected to generate the experiment networks. On both real and artificial, the performance results of proposed algorithm are compared with state of the art algorithms by using various performance evaluation metrics. In particular, the proposed algorithm is proven that it has better accuracy on both real and benchmarks and saves running time as an efficient algorithm.

Table of Contents

3. BACKGROUND THEORY

List of Figures

List of Tables

List of Equations

CHAPTER (1)
INTRODUCTION

The societies and natures are made up of a various diversity of different scale complex systems. These complex systems vary from the Internet to World Wide Web, from stock markets to other economic systems and from power grid to various communication systems. In simple systems, interactions between components are often determined solely by physical distance. However, in complex systems, the interactions between components are often more complicated and may be influenced by a range of factors, including similarity, shared connections, and other factors that can lead to highly correlated one-to-one interactions. These interactions between components in a complex system play a crucial role in determining the function and behavior of the system as a whole. To comprehend the behavior and function of complex systems, therefore, it is necessary to study the pattern of interactions among their components, which is the subject matter of network science. It is the study of complex systems in the form of networks using theories and computer science techniques, mathematics and physics techniques [1]. The network graphs can help as a powerful mathematical tool to study and represent complex systems. For example, in the scientific literature, articles can be represented as nodes in a network, and the relationships between articles can be represented as edges based on the citations between them; the World Wide Web can be represented as a network of linked web pages, with hyperlinks between pages represented as edges in the network; the Internet itself can be represented as a network of routers connected by physical links.

In particular, the widespread adoption of the internet and computer science has led to a significant increase in the number of people who use social networks. Social network is a type of "virtual society" that connects individuals in the real world and facilitates communication, information sharing, and social activities. As a result, social network analysis has become an increasingly important field of research in recent years. The development of graph theory provides a useful framework for studying social networks, as social networks can be represented as graphs. In this framework, each vertex (or node) in the graph represents an entity in the social network, such as a user, a piece of information, or a group. Meanwhile, each edge in the graph represents the relationship between entities, such as friendships, information dissemination, or group membership [2]. A sample diagram of social network is illustrated in figure 1.1. The social network structures have been studied extensively under the notions of sub graphs, network modules, and communities. In social networks and other real networks, the most commonly found features are community structures. Communities or modules are often considered to be the building block of real-world networks, as they can play a crucial role in determining the functionality and behavior of the system as a whole.

Clustering technology has been playing a key role in various real applications including rotating machinery, image processing [3], biology [4], market segmentation [5], and web mining. Clustering techniques and community detection methods both aim to identify groups of objects within a dataset that share similar characteristics or patterns of connectivity. In the context of complex networks, groups of nodes that are more densely connected to each other than to nodes outside the group. That groups are defined as modules or communities, and can be thought of as clusters of nodes that have a high degree of interconnectivity. The traditional clustering algorithms aim to assign each object to a single cluster, such that each cluster is mutually exclusive and there is no overlap between them. If a network consists of several distinct sub-groups that are only sparsely connected to each other, traditional clustering methods may be able to identify these sub-groups as separate clusters. However, in many real-world networks, the boundaries between communities are not well-defined and may be fuzzy or overlapping. In these cases, traditional clustering methods may not be sufficient to identify the underlying community structure.

Figure 1.1 Social network

Researchers have developed various methods to discover communities from the SN such as Lancichinetti and Fortunato [6], Leskovec, Lang and Mahoney [7]. Moreover Walktrap [8] uses random walks to identify densely connected regions of a graph, which correspond to communities. Other traditional methods use a bottom-up approach that starts with individual nodes and progressively groups them together based on some criterion, such as similarity or connectivity. One popular method that used this approach is the FastGreedy algorithm [9], which is based on maximizing an objective function, such as modularity, to identify communities. Other fast algorithm, infomap [10] is a graph clustering algorithm capable of achieving high-quality communities.

However, many of the studies within community detection have been on identifying disjoint or none overlapping communities. This type of detection suggests that the network can be partitioned into densely connected regions in which nodes have more connections to each other than to the rest of the network. But, it has been well understood that people tend to have multiple community memberships because they have different interests, hobbies, or social circles. For example, someone may be part of a sports team, a book club, and a professional organization, each of which represents a different community; On the Internet, individuals can also simultaneously subscribe to or participate in an arbitrary number of groups, forums. This also appears in other complex networks such as biological networks, where a node might have multiple functions. Therefore, overlapped community detection algorithms have been investigated. These algorithms aim to detect a *cover*, which is defined as a set of clusters in which each node belongs to at least one cluster. The authors, Kelley et al. [11] and Reid et al. [12], showed that many real-world social networks have a significant degree of overlap among individuals in the network. For this reason, there is an increasing interest in algorithms for detecting overlapping communities within networks.

The researchers modeled overlapping community detection algorithms by using various strategies to cluster objects or partition a graph. In this dissertation, local expansion strategy is used to detect overlapped objects and communities by clustering objects. Most research have been done for identifying overlapped structures according to this strategy. To uncover both overlapping and hierarchical community structure in complex networks, a pioneering attempt was made by Lancichinetti et al. [13] , Shen et al., 2009 EAGLE (agglomerative hierarchical clustering based on maximal clique) [14] and Wang Min et al. [15]. These algorithms couldn’t find community structures in complex networks and random seed selection. Therefore, proposed an algorithm with lower time complexity for a large scale complex network in 2014 [16]. Belfin [17] designed a model to locate suitable superior seed set by using blended strategy. However, there are still remaining problems of tuning parameters.

In this research, the system is designed to identify overlapping objects based on local community expansion strategy. This system uses extended jaccard similarity to find seed or core nodes and fitness quality evaluation function to extend communities. The less accurate community structures are occurred due random seed selection of local expansion strategy. Moreover, unstable communities are accepted for various parameters which yield different implementation results because significantly depends on parameter setting. Therefore, this research proposes extended jaccard index or similarity to locate good seeds and propose a formula that controls resolution parameters of fitness function in extending communities to avoid different implantation results.

This system firstly chooses the seed among nodes of the whole network based on extended jaccard similarity. The initial community is formed with neighbor nodes around seed. For extending initial community, then, the neighbor nodes are added to the initial community according to their greater fitness value based on fitness quality evaluation function. Then local communities which communities forming with each seed surrounding neighbors are obtained. Finally, overlapping nodes and communities are detected by merging local communities. This system is compared to other overlapped community detection algorithms to verify for better accuracy and saving execution time on different datasets from mostly Stanford network dataset collection. In addition, overlapping LFR artificial benchmarks [18] proposed by Lancichinetti and Fortunato are also selected to generate networks in this experiment. The proposed algorithm well performs on not only real world network datasets but also benchmark datasets. This work is implemented using Apache Netbean 12.5 on java platform with RAM 8GB, CPU @ 2.50GHz.

1.1 Problem Statements

The graph partitioning and community detection problems involve partitioning a network into groups or clusters, but they differ in their objectives and assumptions. Graph partitioning typically involves dividing a graph into a predetermined number of disjoint subsets or clusters. Unlike graph partitioning, in community detection, the number of communities or clusters is not usually predetermined or given as part of the input. The most community detection algorithms and clustering methods need prior knowledge such as thresholds. In these situations, unsuitable threshold value causes decreasing accuracy of community structures. The local expansion methods are effective strategies for detecting overlapping communities within networks. In general, these methods starts with a set of seed nodes, which are selected randomly and seed nodes are then iteratively expanded into communities by adding new nodes that meet certain criteria. As the problem statements,

1) These randomness can bring computational efficiency but leads to low quality community structures.
2) The principled methods to choose the seeds are few and far between.
3) When they exist, they are usually computationally expensive (e.g. using maximal cliques as seeds [19]).
4) In that cases, not only community structures but also overlapped vertices are detected wrongly if chosen seeds are not appropriate.
5) What's more, methods the local expansion algorithms used require parameter setting in quality evaluation functions to decide if vertices should be added or not to the clusters.
6) Those functions rely on their parameters significantly.
7) Therefore, multiple implementations are done by tuning parameters and different implementation results are occurred.
8) That problem fails to provide efficiency in reasonable execution time and causes stability deficiencies in overlapping community detection.

Nevertheless, even many approaches are proposed by researchers to select suitable seed nodes, still have weakness in applied approaches for extending community.

1.2 Motivation of Research

In recent decade, Graph mining is a fundamental research area in artificial intelligence and community detection has been trending as a hot topic in graph mining area. Community detection techniques group vertices into communities and it is also known as clustering that can be thought of as a way to group nodes in a graph based on their similarity, and it can help reveal underlying patterns and structures in the graph. This revealed knowledge can be applied to provide further tasks such as node profile construction or link prediction. For example, detecting influencers or groups of specific product lover in commercial sectors, can advertise targeted users. In addition, discovering researcher communities can refer future potential collaborator in scholarly networks, or recommend relevant papers or research topics to targeted users. In the healthcare sector, community detection can be used to infer possible protein-protein binding relationships. Therefore, community detection tasks a wide range of practical applications across different domains, and it can be a powerful tool for uncovering hidden patterns and relationships within complex networks.

The real situation of real world network like social network is that each object may belong to more than one community. As regards overlapping communities, objects having different roles in forming a community. Finding out the overlapping communities in social networks is an indispensable task. Some methods that adopt both global and local information to detect community structure have been studied. The global methods become computationally infeasible for very large networks or lack of integrity because it requires the typology information of entire network (i.e. there are too many nodes and edges in the network) and making it difficult to apply them to the entire network. Moreover, algorithms required global knowledge of the network are very high the time complexity and space consuming when analyzing large scale networks.

The local community refers to the community that includes a particular starting node, and the nodes within the local community are closely interconnected, while their connections to nodes outside the local community are infrequent. At times, may only be interested in the local community to which the given starting node belongs. For instance, in the co-purchasing product network, may only focus on the community that includes the given product. After obtaining the local community C that encompasses product A and frequently co-purchased products, other products in C can be recommended to customers who have purchased A. In addition, Influencers or core members in a social network are individuals who are highly connected to other members such as followers or friends in online social networks. Thus, they have significant impact and able to spread information quickly. These core members are important for community detection systems like target advertising system. For this reason, identifying good seeds is also an important task for not only local expansion algorithms but also other community detection algorithms. However, the selection of initial node generally affects accuracy of local discovery methods.

As the network structure in practical applications becomes increasingly complex, detecting community structures more accurately and effectively remains a significant challenge that deserves further investigation.

1.3 Objectives of Research

The aim of this research is to provide the more effective algorithm than other overlapped algorithms on same datasets and also address the problem of developing high quality community detection algorithms that scale with large graph datasets. The main objectives are:

- To introduce efficient and effective overlapped community detection algorithm in application area of real world.
- To do the system well for large networks by local expansion strategy that requires only local information
- To avoid inaccurate local community structures because random seed selection
- To detect suitable seed or core nodes by proposed similarity metric
- To avoid stable deficiency of community structures and detected overlapped elements due to resolution controlling parameters.
- To explore the stable community structures by optimized fitness evaluation function
- To improve the efficiency of proposed overlapped detection algorithm with appropriate running time.

1.4 Contributions of Research

The following are the contributions:

- Propose an effective local expansion approach for uncovering overlapped structures.
- Propose seeding method which extended jaccard similarity in identifying core members
- Optimize f fitness function to overcome the problem of resolution controlling parameters when expanding the community
- Propose a formula to control resolution parameter of fitness function f
- Provide the system which detects overlapped community structures in least execution time

1.5 Organization of Research

This dissertation is organized with six chapters including introduction. The introduction describes that community detection becomes increasingly interested research filed and includes problem statements, motivation, objectives and contributions of this research. The remaining chapters are organized as follows:

Chapter 2 provides a comprehensive review of the most frequently published research papers by researchers in the last decade. This chapter divides into sections to explain clustering algorithms. Firstly traditional clustering algorithms and community detection algorithms according to the other literatures are described. The next presents overlapped community detection algorithms.

As background theory, chapter 3 first introduces the definitions of community, community detection, graph types, application research areas, and challenges facing community detection research area. The overlapped community detection methodologies, local expansion strategies, community quality evaluation functions and similarity metrics between nodes are presented in this chapter.

Chapter 4 follows the proposed system architecture and design. It also provides the proposed system, algorithm and methodology.

Chapter 5 describes performance results of this system and proves better accuracy and less execution times by comparing with other overlapped detection algorithms in where system implementation. In this implementation, performance of the system is measured by various evaluation metrics on different datasets.

As summary, it describes overall of this dissertation in detail and discusses future researches in chapter 6. Then, it points out current limitations and suggest some ways of overcoming these limitations through future research.

CHAPTER (2)
LITERATURE REVIEW

In the last decades, several literatures devoted to the discovering the community structure from networks. This chapter will review traditional clustering techniques from literatures focused on detection of non-overlapped communities. The broadly classification of community detection methods to detect disjoint communities and overlapped communities are also discussed. The clustering techniques are reviewed in [20].

2.1 Non Overlapping Community Detection Techniques

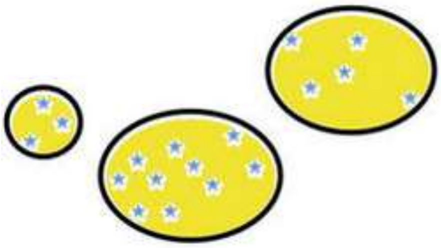


Figure 2.1 Example of partitional clustering

2.1.1 Partitional Clustering

While statistics and data mining have been used in computer science to analyze and cluster data, the first studies on finding communities of similar objects in networks actually originated from social sciences. The most prominent of these old studies are k-means clustering; Partial clustering methods such as neural network clustering and multidimensional scaling are used. The figure of partitional clustering is shown in Figure 2.1. When the edges of the graph are weighted, these properties can be adapted to take these weights into account. Depending on what they represent, these weights can be used as distance measures. The mass consumption in ways of using traditional distance measurements such as K-Means algorithm proves effective in physical networks, especially in the map. But, when the edges have no weight in a graph like graph consisting friend’s relationships, it is inconvenient method to define an appropriate distance and the number of cluster must be predefined to find the clusters.

Therefore, applying traditional distance-measurements are inconvenient ways for social networks analysis. The processing steps of K- Mean algorithm are: Initially communities' size (number of community) is given. In first step, centroid must be specified and the proposed community number on the network keep their numbers as far as possible from the centers of the communities. Each node in the network is assigned to the nearest center. Secondly, the centroids are updated by computing the center of mass or centroids of all the nodes within each cluster. After the centroids have been updated, the algorithm repeats the assignment step, where each node is assigned to the nearest centroid. This process of updating the centroids and re-assigning nodes to clusters is repeated iteratively until the locations of the centroids remain unchanged.The solution may not be optimal and local optimum depends on the initial choice of centroids.

2.1.2 Hierarchical Clustering

Hierarchical clustering is a widely used technique for community detection in social networks. Here is a general description of how hierarchical clustering works for a graph with N vertices and similarity matrix A: [21]

1) Assign all N vertices to distinct community number so this would result in N communities for all N vertices.
2) Locate the two communities that are closest to each other, and then merge them into a single community.
3) Re compute the distance or similarities between the new and all other communities.
4) Repeat steps 3-4 until all vertices are in the same community.
5) The resulting hierarchical tree, called dendrogram, can be cut at a certain level to obtain a desired number of clusters.

Hierarchical clustering methods can be classified into two main categories: divisive methods and agglomerative methods.

2.1.2.1 Agglomerative Method

Agglomerative clustering is the most commonly used type of hierarchical clustering, where the algorithm starts by considering each data point as a separate cluster, and iteratively merges the closest pairs of clusters based on some similarity measure. The result is dendrogram, a tree-based representation of the objects. The process of hierarchical clustering is performed in a "bottom-up" manner. That is, the dendrogram starts with each object being its own cluster (leaf), and then progressively merges clusters together to form larger clusters (nodes) until all objects are grouped together in one big cluster (root).

2.1.2.2 Divisive Method

The divisive method is an inverse of agglomerative clustering, which is also known as DIANA (*Divise Analysis*) , works in a "top-down" manner, where the clustering process starts with a single big cluster that contains all objects, and then the cluster is recursively divided into smaller clusters until each object is in its own cluster. In both the divisive and the agglomerative methods, the final result of merging process generates a dendrogram which represents the merging sequence of communities. To obtain a partition of network, cut the dendrogram at any level. In this network partition, all partition components are viewed as the final communities. See in Figure 2.2.

The hierarchical clustering methods face challenges in choosing the appropriate measurement to determine which communities should be merged in agglomerative or separated in divisive. These two different choices lead to different hierarchical methods to detect community structures. In [22], Girvan and Newman used a measurement based on edge betweenness in a divisive method. They work by using information about edge betweenness to detect community peripheries not emphasize on strong connected core of communities. To solve the expensive computational cost of edge betweenness in divisive methods, furthermore, the edge-clustering coefficient as the measurement was proposed by Radicchi et al. That paper introduced a divisive algorithm based on local quantities, which they called the "fast greedy algorithm". Fortunato et al. also introduced the use of information centrality as a measure for agglomerative hierarchical clustering methods. That work consists in finding and removing iteratively the edge with the highest information centrality. The next weakness is that they did not know where to cut the dendogram tree produced by hierarchical clustering methods. It needs to choose the appropriate location where to cut this dendogram. To address this issue, need a metric to measure quality of goodness of a network partition. When Newman and Girvan study their different methodologies based on edge betweenness, proposed modularity [23] which is a well-known measurement. The appearance of modularity has been a driving force in the development of community detection methods. The development of modularity-based methods has led to a rich variety of optimization techniques for identifying community structures.

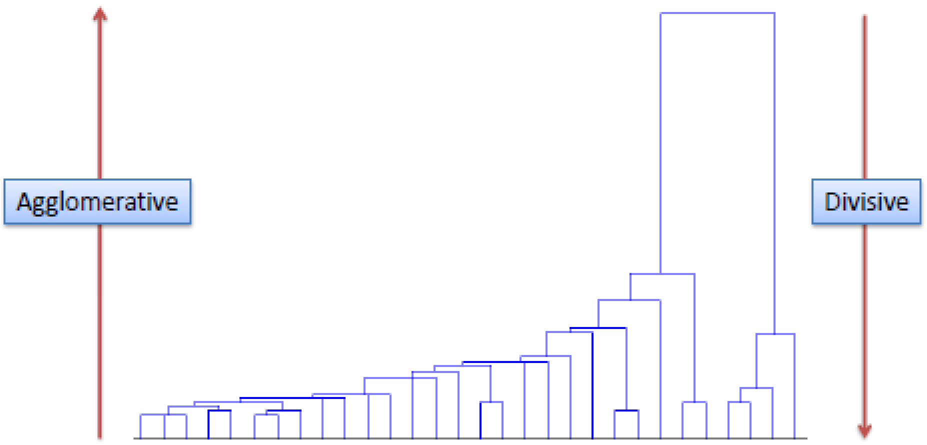


Figure 2.2 Hierarchical clustering [21]

2.1.3 Modularity Optimization

Optimization techniques are widely used in clustering to find the best possible partition or clustering of data. The quality of a clustering is often measured using a quality function or objective function that quantifies how well the clustering captures the underlying structure of the data. The goal of optimization is to find the clustering that maximizes or minimizes this objective function.

The modularity function is one of the most popular quality functions used in community detection or clustering of networks. The modularity function assesses the effectiveness of dividing the network into communities by measuring its quality. Therefore the modularity measure difference between the original network structures and randomized version of the original graph. By comparing the actual network structure to its randomization, modularity reveals how non-random the group structure is.

By revising the referenced null model with different constraints, the modularity has been extended to different type of networks such as weighted networks [8], directed networks, bipartite networks and multiplex networks. With the modularity, the community structure of a network can be identified by finding the optimal partition that maximizes the modularity. The process of finding the optimal partition involves exploring the space of all possible partitions of the network. This optimal partition represents the grouping of nodes into communities. Modularity maximization is NP-hard. Therefore one can realistically hope to find only suitable approximations of the modularity maxima and several distinct approaches has been proposed. Indeed, after it was introduced, it appeared to capture the fundamental nature of the problem and provide the solution's key. However, The problem such high modularity due to randomness fluctuation of random network generated by configuration model and resolution limit problems further affect the practical performance of modularity.

2.1.4 Graph Partition

Graph Partitioning is a typical process in computer science. The graph partitioning approach separates the network into groups of equal size while attempting to minimize the number of links between them. Many graph partitioning methods, including the Kernighan Lin algorithm and the spectral bisection method, are based on iteratively dividing a graph into two separate groups. The Kernighan Lin algorithm is a heuristic algorithm that starts with an initial partition of the graph into two subsets and then iteratively swaps nodes between the two subsets in a greedy manner to improve a given objective function. The spectral bisection method, on the other hand, uses the eigenvectors and eigenvalues of the Laplace matrix of the graph to partition the graph. The Kernighan Lin algorithm [24] is a specialized approach to spectral bisection, which is another graph partitioning algorithm that is based on the spectral properties of the Laplacian matrix of the graph. It tries to maximize the benefit function. The benefit of the function is the difference between sum of the weights of the edges within the sets and the sum of the weights of the edges between the sets.

Its primary drawback is that it requires the community sizes to be predetermined during the initial phase. It is inconvenient for real world datasets as the results heavily depends on the initial size and configurations. Later, [25] is extended that is not required to specify the number and sizes of communities. In that extended form, a single node moved to other communities at a time but it also has disadvantages such as timing and poor detection of communities. Therefore, one of the drawbacks of graph partitioning methods is that they usually require to choose the number of partitions. To address this issue, use modularity that is a goodness metric for evaluating the partition of the graph at each step. However, this is computationally expensive and cannot be feasible for large graphs.

2.1.5 Genetic Algorithms

Genetic Algorithms are a type of optimization algorithm which mimics the science of genetic and natural selection. In real world, individuals’ crossover their genes in which their genetic information are hold, and produces new offsprings. In addition, sometimes, a gene of an offspring can be changed or mutated. If the offspring's mutated chromosomes and crossover have favorable genetic traits that allow them to adapt to their environment, they will survive. Individuals who cannot adapt the environment will become extinct. In genetic algorithms, better solutions can be constructed by leveraging the best partial solutions from previous iterations rather than exhaustively searching the entire solution space. This concept is referred to as the building block hypothesis. Genetic algorithms employ various techniques, such as crossover, selection, and mutation operators, to generate optimal solutions from an initial set of solutions. A fitness function is used to evaluate an individual's score, which measures how well the individual fits the solution or the environment. After the evaluation phase, offsprings with best fitness score survive for next generations. The following steps are process of Genetic Algorithm [26]:

1) Initially, a predetermined number of chromosomes are created, which is also known as the population size. Every chromosome is assessed using a cost function, which is commonly referred to as the fitness function.
2) The GA generates improved genetic sequences called chromosomes, which then replace weaker ones in order to maintain a healthy population. This process is known as selection.
3) The process of crossover produces new chromosomes with genetic material that is a combination of two parent chromosomes.
4) If a given probability met then some of the chromosomes are impacted by a mutation.
5) If a given probability met, then certain chromosomes are impacted by an inversion.
6) Every newly created chromosome is evaluated and adjusted according to a fitness function.

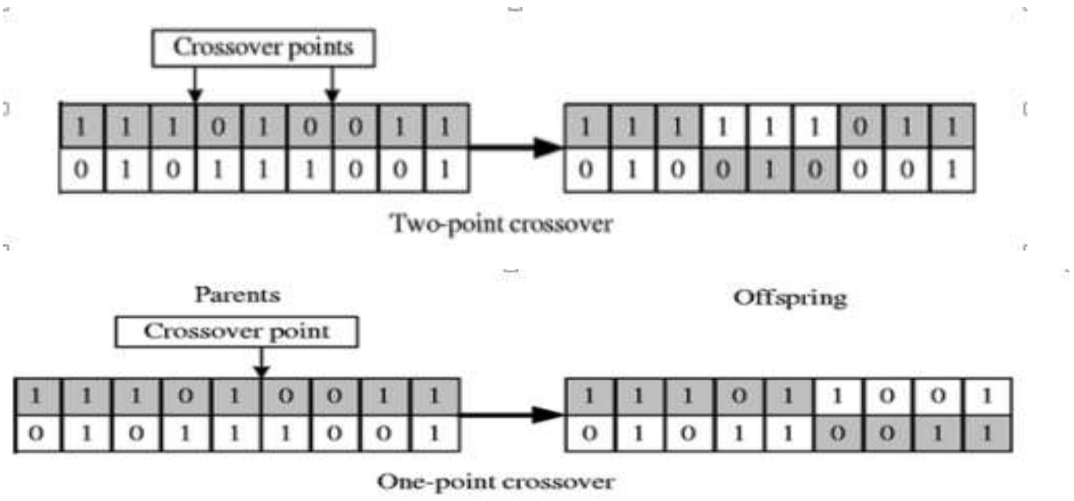
7) Go to step 2 until reaches aimed fitness score.

After applying the fitness function to each chromosome, the selection step involves choosing a predetermined number of chromosomes from the solution space, which will be utilized in the next iteration of the algorithm.

The primary process of the genetic algorithm is crossover, in which two chromosomes are randomly selected from the solution space and determine one or multiple crossover points on each chromosome, depending on the type of crossover selected. In one-point crossover, a point is selected on each chromosome and the chromosomes are then divided into two pieces. The pieces with the same location and size on both chromosomes are swapped with each other, creating new offspring chromosomes. In two-point crossover, chromosomes are divided into two sections by selecting two points along the chromosome. The middle sections of the chromosomes are then swapped with each other, creating new offspring chromosomes.

In uniform crossover, certain genes of one parent's chromosome are swapped with the genes of the other parent's chromosome. These specific gene swaps are known as crossover bits. These crossover operation is shown in Figure 2.3.

Mutation is a random process in which a selected chromosome undergoes changes in the values of its genes. The occurrence of mutation is determined by a predefined probability function. When this probability is met, one or more genes in the chromosome are randomly selected and their values are altered. This change can affect a single gene or multiple genes at once. In Fig. 2.4 and 2.5, a mutation and inversions example is shown. Inversion operator is not always used in real datasets. It does not change the genetic information, it changes the presentation of these genetic information.



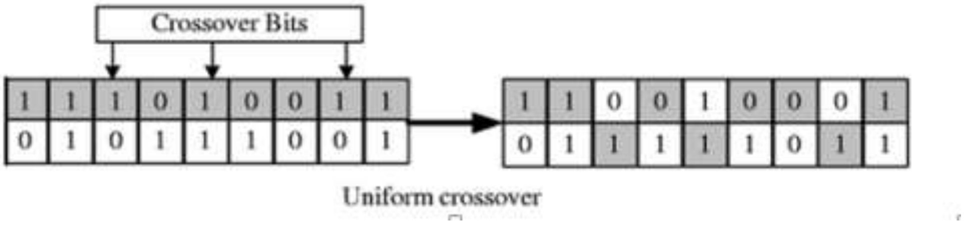


Figure 2.3 Crossover in genetic algorithm

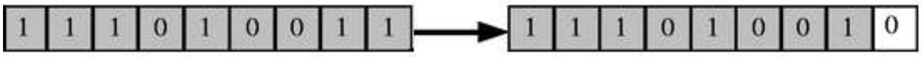


Figure 2.4 Mutation in genetic algorithm

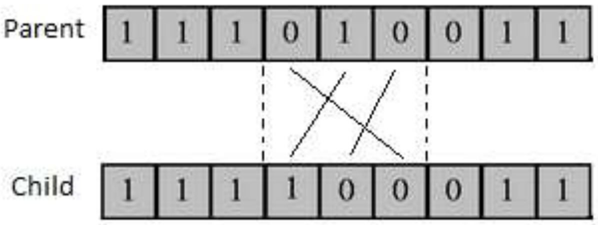


Figure 2.5 Inversion in genetic algorithm [27]

2.1.6 Spectral Clustering

This clustering method uses the spectral properties of a graph to identify clusters. The eigenvalue spectrum of various graph matrices, such as the Laplacian [28] and adjacency matrices, are analyzed to detect densely populated eigenvalues that are closely spaced, along with a few outlying eigenvalues that deviate significantly from the cluster. The information of the large-complex network structure, like community structure can be found in the eigenvectors [29] corresponding to these outliers. Spectral clustering is a technique that involves representing graphs in a matrix space and using the entries of eigenvectors as coordinates. Each element of the eigenvectors represents the coordinates of a vertex in a k-dimensional Euclidean space, where k is the number of eigenvectors used. The standard partitional clustering techniques like *k*-means can group into clusters the resulting points. However, Spectral clustering is not always reliable because the distinction between the eigenvalues of the eigenvectors of related community when the network is very sparse and the bulk is not sharp. Eigenvectors that correspond to eigenvalues beyond the main bulk of values may be linked to nodes with a high degree of connectivity, also known as hubs, instead of being related to the structure of groups.

Similarly, Eigenvectors that are related to communities tend to have eigenvalues that fall within the main bulk of values. When selecting eigenvectors based on whether their related eigenvalues are inside or outside of this range, a diverse set of information can be obtained that includes both community structure and other features like hubs. Using those eigenvectors makes community detection more difficult, sometimes impossible for the spectral clustering procedure. Unfortunately, many of the networks corresponding to the studies encountered are very sparse networks and can lead to this type of anomalies. Standard matrices (Laplacian, adjacency matrix, modularity matrix, etc.) indeed fail in sparse networks built with spectral methods from planted partition models before the theoretical detection limit.

2.1.7 Statical Inference based Methods

To tackle the community detection problems, statistical inference provides a powerful set of tools and is a standard approach to fit a generative network model on the data. The stochastic block model (SBM) [30] is by far the most widely used generative model of graphs with communities. The typical SBM (Stochastic Block Model) generated the highest possible log likelihood (not normalized) for a specific partition of the network into q groups. Although the model mentioned does not consider the variations in the degree of nodes that exist in most actual networks, resulting in an inaccurate depiction of the group structure in many of these networks. As a solution, Karrer and Newman suggested the use of the degree-corrected stochastic block model (DCSBM) [31]. This model kept constant degrees of vertices on average with introduced of additional suitable parameters. The main constraint of this method is that it requires the specification of the number of groups, denoted by q, before analysis, and this value is often not known for real-world networks. The model selection is possible action and it’s the best if can choose the model that best compresses the data. The degree of compression can be approximated by considering the total quantity of information required to characterize not only the adapted model but also the information required to characterize the model itself, which increases as a function of the number of blocks q.

A nested hierarchy of stochastic block models consists of more refined model selection methods and an upper level of these hierarchy serves as prior information to a lower level. This study lowers the resolution boundary to log n, allowing for the identification of smaller blocks that were previously undetectable. Other methods have also been suggested to determine the number of groups. Other techniques to extract the number of groups have been proposed.

2.1.8 Dynamic Algorithms

This section describes the methods that apply processes running on the graph, emphasizing on spin-spin interactions, random walks.

Spin models: The Potts model is one of the most popular models in statistical mechanics. It presents a spins system that can exist in different states. With this idea, Reichardt and Bornholdt developed a method to find communities that maps the graph onto a zero temperature q-Potts model with nearest-neighbor interactions. In another work, Son et al introduced a clustering technique based on the Ferromagnetic Random Field Ising Model (FRFIM) [32].

Random walk [33]: Random walks can also be used to detect communities. If a graph has a strong community structure, a random walker spends a long time within a community due to the highly dense interior edges and large number of paths it can follow. Zhou measured a distance between pairs of vertices by employing random walk. The distance d_{ij} between i and j from this measurement is the average number of edges that a random walker has to cross to reach j starting from i. Later, Latapy and Pons proposed a different distance measurement between vertices based on random walks. In this proposed measurement, the distance is calculated from the probabilities that the random walker moves from a vertex to another in a fixed number of steps. A graph clustering technique is designed by Hu et al and it is based on a signaling process between vertices, somewhat resembling diffusion.

2.2 Overlapping Community Detection

There is a class of clustering algorithms for graph clustering known as overlapping community detection algorithms that allow nodes belonging to more than one community. In recent years, many researchers have studied algorithms to detect overlapping structures in networks. The algorithms are categorized into four approaches: Clique Percolation, Link Partition, Local Expansion, and Label

Propagation. In addition, fuzzy detection and overlapped detection on dynamic networks have been studied by researchers.

2.2.1 Clique Percolation Method

The Clique Percolation Method (CPM) [34] is a popular algorithm for identifying overlapping communities in complex networks. It begins by creating the vertices of the k-clique graph and then constructing the edges of the graph (percolating when two vertices in the k-clique graph have strong connections). In the resulting graph, each connected component of the clique graph is considered a community.

CFinder [35] is a system to identify overlapping groups of nodes in undirected graphs and can visualize this clusters (groups of nodes) by the system. It also allows navigation of the original graph and the communities found. The search algorithm, CFinder uses the Clique Percolation Method to find k-clique percolation clusters. A k-clique is a complete subgraph composed of k nodes (e.g 3 clique means a sub graph composed of three connected nodes). So, the concept of "adjacent cliques" is adjacent cliques as two cliques that share exactly k-1 nodes, meaning they have k nodes in common. The parameter k represents the size of the cliques, and the authors suggest that a value between 4 and 6 would give the richest group structure. The higher the value of k, the smaller the size of the denser groups.

Greedy Clique Expansion (GCE) [36] has been developed to find accurate overlapping communities and obtain good performance on synthetic data. It is used to obtain the community C with first identifying the seeds (core nodes) by finding the maximum cliques and then add nodes until the inserted nodes have less fitness. A method, ECPM (Extended Clique Percolation Method) [37] is modeled to tackle problem of not discovering complete network in CPM. ECPM, firstly, detects initial communities by identifying nodes that are densely connected to each other. The left-out nodes, which are not included in any initial community, are then assigned belonging coefficients based on their connectivity with the nodes in the initial communities. This allows the algorithm to extend the communities by adding nodes with the highest belonging coefficients. Finally, ECPM merges the most similar communities using Jaccard Similarity. This ensures that the resulting communities are cohesive and well-connected. However, the majority of algorithms that rely on CPM have a high level of complexity as they attempt to identify numerous small cliques in order to define communities.

2.2.2 Link Partition

Some link partition algorithms can detect overlapped community structures easily the whole network. The link partition method typically involves two steps. First, the link graph is constructed by treating the edges as nodes and the original nodes as edges. Then, a graph partition algorithm is used to divide the link graph into communities based on the connectivity between the edges. Alternatively, some link community similarity functions can be used directly for clustering without a graph partition algorithm.

Link clustering techniques are proposed to identify overlapping communities by dividing the set of links rather than the set of nodes. The line graph is used to this end. The primary benefit of utilizing clustering on the line graph is that it generates an overlapping graph of the initial interaction graph, which permits nodes to exist within numerous communities simultaneously. Ahn *et al.* [38] designed LINK and its' basic idea is to divide edges instead of nodes to uncover community structure. According to the given pair of edges on graph, the similarity can be computed based on the Jaccard Index and the edges are partitioned via hierarchical clustering of edge similarity. LINK first performs a hard partition on the set of edges in the network, and then the result is transformed into the right side of the corresponding community structure of nodes. A node in the original graph is called overlapping if edges connected to it are put in more than one edge communities.

A simple and innovative technique [39] is suggested for networks with loose connections that cannot be resolved by CPM. Loosely connected networks yield inaccurate overlapping community structures in weak-tie membership. Therefore, LinkSCAN is a method that involves transforming an original graph into a link-space graph using link-space transformation. It detects overlapped groups by applying using link similarity in original graph and non-overlapping community detection algorithm in line graph. Then they enhanced the efficiency of LinkSCAN by sampling based on this framework. MEME link, link based clustering algorithm [40], is developed which optimizes the modularity density function. It uses a weighted graph and a similarity function to identify densely connected links among communities. In addition link-based algorithms, the researchers also studied line graph theory, ensemble learning, and particle swarm optimization. They found that the conventional algorithm using PSO has a weakness that causes the creation of unnecessary small-sized communities.

Therefore, an efficient algorithm LEPSO (LinE graph Particle Swarm Optimization) [41] is proposed to address these problem.

2.2.3 Local Expansion

The seed expansion method generates overlapping communities by selecting seeds and expanding the seed by using various fitness functions, then combines the intermediate communities into larger, global communities. The challenge with this approach is to identify suitable initial seeds. In this approach, each implementation varies greatly depending on the various fitness functions.

Lancichinetti et al [42] introduced Order Statistics Local Optimization Method, called OSLOM. It attempts to identify groups within the network that are statistically significant and it especially was performed in for a random network without community structure. The statistical significance of a community is determined by the likelihood of discovering a community that possesses similar characteristics. To detect communities, nodes are added or removed from the community while estimating its statistical significance. The maximum time complexity in the worst-case scenario is O (n^2), where n represents the number of nodes.

Lee et al. [36] developed GCE, Greedy Clique Expansion and it first begins by taking maximal cliques as a seed set. They are then expanded by greedily maximizing a local fitness function. Finally a check is performed to eliminate the nearly-overlap cliques and communities. A Clique Coverage Heuristic (CCH) is used to remove nearly-overlap of cliques. Each clique with more than φ proportion of its nodes is removed if it belongs to at least two already accepted large cliques. Also, all seeds within a small distance of an already accepted community are discarded.

In [43], a seed set expansion algorithm is presented to search overlap communities. Although the seed set expansion algorithm is one of the most elegant overlapping community detection algorithms, the algorithm is computationally expensive for detecting the community structure in the network community analysis using maximal cliques as seeds. This work is performed by local expansion and optimization based on growth natural community. Selection of seed kernel is based on the distance as the *k*-means and spectral clustering and seed selection functions can determine the good seed on a node in the network. In particular, this seed set expansion algorithm proposes a strategy involving many computing clusters using a weighted kernel *k*-means algorithm on the graph. This algorithm consists of filtering, seeding, seed set expansion, and propagation phase. The communities' quality measurements such as cut, conductance and normalized cut are applied to measure the cluster quality in those work.

Cut. The "cuts" of cluster Ci are determined by calculating the total weight of the edges between Ci and its complement, V \ Ci. In other words, the cut (Ci) is equal to the sum of the links between Ci and V \ Ci.

Conductance. The conductance of a cluster is defined to be the cut divided by the least number of edges incident on either set C_i or $V \backslash C_i$. By definition, cond(C_i) = cond($V \backslash C_i$).

Normalized cut. The normalized cut of a cluster is defined by the cut with volume normalization. Notice that ncut (C_i) is always lesser than or equal to cond (C_i). In 2016 [44], a local community was identified by first detecting a minimal cluster using the density function, and then identifying nodes that are closely connected to the initial nodes. To find a good seed, Overlapped Community Detection Node-Weighting was designed by Chen and Li [45]. The algorithm has three phases: seed selection, community expansion and community optimization. In seed selection phase, identifies the seed with the highest weight of each node by the total score of all edges connected to its neighbors. Then the seed is expanded by adding neighbor nodes by using node fitness function to form a local community. Finally, if two communities have a significant amount of overlapping vertices, they are merged together to form larger clusters in order to enhance the overall quality of the community.

2.2.4 Label Propagation Method

Label propagation method refers to labels being propagated between nodes. It assigns unique label to each vertex and propagates these labels through the whole graph. When updating labels; if a vertex receive multiple labels, connections between them are broken randomly. But label propagation algorithms can only uncover disjoint community.

COPRA [46] is an iterative method based on the concept of multi-label propagation with computational complexity close to linear. It extends the label propagation algorithm (LPA) [47] with the ability for every node to have multiple labels. The following are the steps involved in COPRA:

(a) Initialization: assigns a unique label to each node in the network, C_i (0) = I and initialize its belonging coefficient as 1, (C_x, l).

(b) Iterative label propagation: each node updates its label and corresponding belonging coefficient based on its neighboring nodes.

(c) Community division: divide all nodes that share the same label into a community.

COPRA algorithm inherits some shortcomings of the original label propagation algorithm and can find the unstable community structure. One of the many drawbacks of COPRA is mostly related to the number of fixed communities’ *v* where is a parameter of the algorithm. A BMLPA [48] method was proposed to avoid this problem, but the researchers did not provide an implementation, which made it difficult to compare with this method. Another weakness of the high-variance COPRA algorithm is its non-deterministic nature, which makes interpreting the results challenging. The randomness in the algorithm mainly stems from two factors. The first factor is that the initial labels are assigned randomly. The second factor is part of the label propagation process, where if multiple labels have the same maximum belonging coefficient below the threshold, COPRA retains one of them randomly.

To avoid detecting only disjoint community, SLPA [49] is developed and it can find overlapped communities because receives multiple label. The algorithm starts with each node in the network being assigned a unique label. Then, for each iteration of the algorithm, a random listener node is selected, and its neighbors send their labels to it. The listener node then chooses one of the received labels based on a listener rule and adds it to its set of labels. In last process, a threshold is utilized for post-processing to construct overlapping communities. Some LPA-based algorithms require setting parameters as prior knowledge and consequently, instability of community structure can be faced and less accuracy. In 2018, Label Propagation Algorithm with Neighbor Node Influence (LPANNI) [50] is developed to overcome that weakness. When propagate the label of it, use label update strategy that considers both the influence of neighbor nodes and the historical preferences of each node for certain labels. The label update strategy combines ideas from COPRA and DLPA which can detect overlap and reduce complexity, respectively. LPANNI calculates the node importance and order by ascending to find core node and then it detects overlapping community structures by adopting fixed label propagation sequence corresponding node importance.

2.2.5 Dynamic Networks

The aforementioned community detection efforts are mainly focused on the community structure of a static network. Most real world social networks are fundamentally dynamic and rapidly growing interaction among the social media. Community structures evolve over time and network communities are also very dynamic. But, there has been little research on analyzing these dynamic communities. This is due to two main reasons. Firstly, it is still unclear how to identify the underlying communities in static networks. Secondly, it is difficult to obtain network data with timestamps. Recent methods also aimed at finding static dynamic communities are described.

Anita Zakrzewska [51] presented a dynamic seed set expansion algorithm, which identifies communities within a network that may change over time. By incrementally updating the community as the underlying graph changes, the algorithm can maintain a local community and adapt to new information without having to recompute the entire community structure. That algorithm focuses on finding local community via given seed set and this dynamic algorithm produces high-quality communities that are similar to those found when using a standard static algorithm.

An extension algorithm of SLPA, SLPA Dynamic (SLPAD) [52] was improved and it can handle dynamic networks. The basic idea behind SLPAD is to update the community structure of SLPA at each timestamp based on the edge changes that have occurred since the previous timestamp. Specifically, for each new edge, the algorithm updates the corresponding node's memory and recalculates the node's label by considering the labels of its neighbouring nodes. However, SLPAD only takes into account updates that involve changes to the edges in a network and not changes to the node [53]. In 2018, a detection strategy that uses agents to identify overlaps was modelled. This strategy examines the network and updates the communities accordingly in response to any changes in the network. In order to detect communities, iterative assignments technique is used for assigning each node to communities based on attribute similarity. One of the strengths of this approach is that it enables all types of community operations, such as birth, death, growth, contraction, merge, and split. OLCPM (OnLine Clique Percolation) [54] is a hybrid technique and it is designed on idea of both label propagation and clique percolation methods. That research presented online version of the CPM algorithm (called OCPM) that builds on the original CPM to detect the core nodes of communities in real time and aims to analyse the network's dynamic behaviour that may result from insertion or removal of nodes. As post-processing based label propagation, OLCPM is defined in OCPM's generated communities to discover the peripheral nodes.

Most of the existing work that deals with the detection of dynamic communities focuses on snapshots of networks. This approach may not capture the mechanisms of community evolution and may be unable to accurately predict the evolution of communities. In cyberspace, the evolution of explicit communities provides us with valuable data to analyze the mechanisms of community evolution and design algorithms that can accurately estimate the evolution of these virtual communities.

2.2.6 Fuzzy Detection

Fuzzy detection approach evaluates the association strength between all pairs of nodes and communities. Ding et al. [55] expanded on the correlation-propagation clustering algorithm to identify overlapped communities and developed a method. This method uses representative exemplars to identify the clusters. First, the nodes are represented as data points in the Euclidean space using the commute time kernel, which is a function of the inverse Laplacian. The cosine distance is applied to measure similarity between nodes is then measured by the cosine distance. In SSDE (Sampled Spectral Distance Embedding) [56], the network is initially transformed into a d-dimensional space using spectral clustering. Next, a Gaussian Mixture Model (GMM) trained using Expectation-Maximization algorithm. The number of communities is determined when log-likelihood increase of adding a cluster is not significantly higher than adding a cluster to random data around the same space.

Non-negative Matrix Factorization (NMF) is a technique used in machine learning for extracting features and reducing dimensionality, which has also been applied to community detection. Zhang et al. [57] substituted the feature vector commonly employed in Non-negative Matrix Factorization (NMF) with the diffusion kernel, which is a mathematical function derived from the Laplacian of the network. Later Zarei et al. [58] showed that defining the matrix with the correlation matrix of Laplican columns gives the better results. Recently, Yang and Leskovec [59] developed BIGCLAM (Cluster Affiliation Model for Big Networks) based on the Non-negative Matrix Factorisation approach. McDavid et al. [60] introduced Model-based

Overlapping Seed Expansion (MOSES) and it is a fuzzy-based detection algorithm. MOSES constructs a modified OSBM (Overlapping Stochastic Block Modeling). Initially, the edges are chosen randomly and each edge represents an initial community. After that, by optimizing a global fitness function, they are greedily expanding communities. Finally it periodically removes the entire community to improve the fitness function. After the expansion of the edges, nodes are eliminated from the communities they belong to and added to different communities to improve the fitness function. The time complexity of MOSES is *O (mn*2*).*

2.3 Related Works

Lancichinetti et al. made significant efforts to uncover both the overlapping structure and hierarchical community structure of complex networks, paving the way for future research in this area [13]. They tried to uncover the overlapping communities in the network by considering the basic concept of the local optimization of a fitness function. The steps of LFM (Local Fitness Method) are as follows:

1) As initialization, pick a node at random as the original member of initial community G_0,
2) The fitness of all neighbor nodes to G_0 are computed according to the f fitness evaluation function.
3) The neighbor with the largest fitness is added to community, yielding a local community G';
4) Recalculate fitness of each node in G'.
5) If a node leads to negative fitness, it is removed from G_1, yield a new community G".
6) If step (5) is occurred, repeat from step (4), otherwise repeat from step (2).

This process stops when all nodes in step (2) have positive fitness. Then the algorithm continues to select other node not including G_1.

The process is repeated until all nodes in the network have been assigned to at least one community. The approach is able to identify the hierarchical relationship among these overlapping communities around a specific node. The parameter of fitness function is range [0.6-1.6] and they tested their algorithm by tuning the parameter within that range. A remaining problem is that it cannot be guaranteed that the hierarchy of all overlapping communities in the network will be detected, due to the random selection of seed nodes.

Shen et al., 2009 [14] addressed the challenge of detecting overlapping and hierarchical community structures in a graph. They introduced a method called "EAGLE" (agglomerative hierarchical clustering based on maximal clique) that utilizes maximal cliques as fundamental components to identify the community structure of the graph. The presence of intersections among distinct maximal cliques enables the agglomerative hierarchical clustering process to identify the overlaps between communities and the hierarchical relationships among these overlapped communities. That work proposed a measurement EQ called modularity, to evaluate the quality of a cover of network. By directly optimizing the proposed measure, the algorithm can then find the overlapping community structure. What is more, optimizing the new measurement EQ on the original network to identify the overlapped structure is similar to optimizing the standard modularity on the maximal clique network. Therefore, any approach that relies on optimizing modularity can be applied directly to detect the overlapped community structure.

Wang Min et al. [15] improved LFM and EAGLE to solve problems of inefficiently and accurately detection at the same time of before algorithms. It was proposed by combining fitness function optimization and community similarity, which can uncover both overlapping and hierarchical community structure of complex networks. The basic idea is to use fitness function optimization at the bottom of hierarchy division to identify efficiently and accurately the underlying community structure which is with overlaps. However, still remain the problems of random seed selection and couldn’t find community structures for complex networks.

Therefore, for a large scale complex network, proposed an algorithm with lower time complexity and higher classification accuracy for detecting overlapping community structure based on LFM algorithm in 2014 [16]. It applied the fitness function for the local optimization. The first step of this method is to choose maximum degree node and its some special adjacent nodes i.e. fitness function satisfies the condition. After considering as initial community these nodes, expand the initial community by repeatedly adding qualified nodes to it. Then remove those nodes whose node fitness are negative from the community. This algorithm set 1 as parameter value on different datasets.

In 2017 [61], the result of traditional LFM algorithm is full of instability because of randomly selection of seed nodes. What's more, the accuracy of LFM decreases apparently in network with fuzzing community structure. In order to solve the problems, LFMs which improved LFM algorithms is designed and also considered weighted information. It used random walk method to select seed nodes to avoid the instability of LFM. Then, with cosine similarity to calculate vertex similarity, weight information in network was fully used. This work redefined fitness function based on similarity matrix**.** This algorithm is validated on LFR benchmarks and real networks with parameter setting 0.8 to 1.2. But it is found effectively performed on random benchmarks. However, these algorithms tested with various parameters and differs implementation results which depends on parameter setting.

Havemann et al [62] greedily expands natural communities of seeds until the whole graph is covered by using a local fitness function. The analysis of communities' hierarchy is achieved by determining the levels of resolution at which communities develop, instead of relying on numerical values obtained through the implementation of various resolution levels. In this work, As nodes cannot remain isolated when the fitness function defined by Lancichinetti et al [13] is applied, the researchers modified the formula f (G, ∞) by increasing the numerator by 1. This analytical procedure is not only more accurate but also faster than its numerical alternatives such as GCE and LFM. The important resolution levels can be determined by finding intervals where large changes of the resolution don't result to growth of communities.

The local first discovery method, DEMON (Democratic Estimate of Modular Organization of a Network) [63] is a node-centric bottom-up overlapping community search algorithm. It exploits ego-network structures and overlapping label propagation to identify individual-level scale communities that consequently aggregate into group-level ones. Another overlap detection algorithm [42], OSLOM (Order Statistics Local Optimization Method) is based on optimizing the fitness function by local search through adding or removing nodes from the cluster. Each cluster is defined as a single vertex and new vertices are joined if the corresponding clusters have common edges. The weights of the new edges are set proportional to the number of the edges between the original clusters. As summarize, firstly find clusters via local search to maximize the fitness function and iterate until convergence. Then group or split clusters based on their internal structure. Finally consider clusters as vertices and iteratively build a hierarchical structure of clusters. It can work on small networks as well as large networks. However, the limitation of this method is occurrence of homeless nodes.

A method of clique-based, K-clique percolation [64] is an overlapping community discovery algorithm that extracts discrete structures consisting of overlapping cliques from complex networks. Some aspects of k-clique percolation are computationally challenging in practice although this is conceptually simple and elegantly expressed using clique graphs. It is developed to address poorly performance on networks with the kind of spreading overlapping community structure that are mostly occurred in many real world social networks. The method considers several computational aspects about the problem of percolating structures in large complex networks. They are still fundamentally limited by the need for cliques to experiment with other cliques sharing certain nodes, but do not percolate. These clique based algorithms occurred NP hard problems and remained problems for large scale network with million nodes and edges.

Therefore, Y. Jaewon developed BigClam [59] and it constructed on models of nodes' affiliation to communities that maximize an objective function using non-negative matrix factorization. Non-negative factor describes the degree of membership of that node to the community and each node-community pair is assigned a non-negative factor. Then, the probability of an edge between two nodes is modeled as a function of shared community ties. The intuition behind this model is that when they share more communities, neighbors are more likely. BigClam set community *c* to -1 the number of communities to detect; *mc* and *xc* to 5 and 100 respectively to find the minimum and maximum number of communities. These are the default values proposed by the researchers. However, this algorithm and algorithms such as link clustering and k-clique percolation required global information of whole network and it is difficult to attain this information for large scale networks. Therefore, local approaches have been increasingly interested in community detection research areas.

In 2015 [36], Y.Xing et.al proposed a novel algorithm based on local approach. Many local community discovering algorithms begin with single vertex and uncover local community corresponding to the vertex by using an optimized function. Then local clusters or communities are integrated according to the integration criteria. But, the algorithms start from any node and cannot get the expected results. Because of this weakness, OCDLCE (Overlapping Community Detection algorithm by Local Community Expansion) was proposed. It starts with each edge of the network and sets the pair of nodes connected to the link as the initial node set. It uses the M function as a standard to obtain local communities by adding nodes and extending the local community to obtain the final structure. The OCDLCE algorithm has three main steps: searching local community, community merging and refining community.

Because of importance of stating node in local discovery approach, the researchers have explored seeding algorithms. Moradi et.al [65] proposed a novel seeding algorithm that is parameter free and utilizes only the local structure of the network to identify good core nodes (seeds) spreading over the entire network. In order to find that seeds, algorithm first calculates similarity indices from local link estimation techniques to assign a *similarity score* to each node, and then uses a *biased graph coloring* algorithm to enhance the seed selection. The proposed biased graph coloring algorithm enhance seed selection because it favors the nodes with high similarity scores. The implementation of this task on large-scale real-world networks demonstrates its ability to choose good seeds, which are then expanded into high-quality overlapping communities that encompass the vast majority of nodes in the network. This is achieved through the use of a personalized PageRank-based community detection algorithm.

R. V. Belfin [66] identified the most important nodes in the community using a parallel superior seed set selection (P4S: degree centrality, engine value centrality, local clustering coefficient and page rank centrality) algorithm. According to $\mathrm{Ex}^{\mathrm{Max}}{}_{\tau}$, the maximum expansion limit t, he identified superior seeds would be expanded by their neighborhood till it reaches the next seed. The distance matrix is applied to compute distance between seeds and the vertices that are not chosen in the first iteration will be taken and added to its neighbors' community. In case of more than one neighbor, the node with the highest degree will insert its non-member node to its community. As the algorithm extends using the neighborhood, it will form the closely knit group around the seed nodes. The goodness of clusters is evaluated by inter cluster density, intra cluster density and clustering coefficient. The density ρ of graph can be evaluated to attain the cohesiveness of the edges in a graph. The density of the sub graph is induced by the cluster as the internal or inter-cluster density. The outer or inter-cluster density of clustering is defined as the proportion of inter-cluster edges to the maximum number of inter-cluster edges feasible. Moreover, these researchers also tested by applying other various centrality measures and using them to find overlapping communities. That model find the superior seed set by applying the local clustering coefficient, page rank centrality, degree centrality and eigen vector centrality [67]. The algorithm makes costly due to the use of betweenness and closeness centrality. The system used set operators and a threshold value τ to limit the number of seeds. After determine seed set, the next steps are follows:

(a) Find the distance between seeds by preparing distance matrix S (g) dist.

(b) Define the expansion distance Exdist by estimating min(S (g) dist).

(c) The Superior Seeds S (g) will be expanded according to the Exdist hops.

(d) At that time, an initial local community will be formed and some unmapped nodes U (g) also will be comprised in the input network.

(e) Determine a distance matrix U (g) dist to manipulate the distance between S (g) and U (g).

(f) Added Each U (g) nodes to the closest seed by referring the U (g) dist.

(g) Finally, added all nodes from the input graph G to various communities.

Unfortunately, the use of shortest path algorithm in the seed expansion model makes the process expensive for graphs with large scale although convenient for graphs with a small number of edges. Moreover, many overlapping detection algorithms have the problems such as excessive overlap, predefined parameters and no guarantee of stability of multiple runs.

Recently, appeared algorithms conceptualized on neighborhood similarity when expanding communities to identify local communities. Long Chen [68] proposed the method of calculating the node degree of membership in order to address the issue like parameters. . It detects overlapping communities with local optimal expansion cohesion idea. Firstly, the important node is considered on clustering coefficient and builds the initial core community consisting of the highest importance node and its neighbor nodes. Secondly, node attribution degree is used to expand the core community until the termination condition of the algorithm was satisfied. This node attribution degree is designed on jaccard similarity and traditional similarity ignores the direct edge between nodes in this method. To improve quality of community, they took into account both the internal and external links of the community. They evaluated the community's quality by calculating the ratio of the number of edges within the community to the total number of edges, including those inside and outside the community. However, it verified on only small real world networks. Next, overlapped objects detection algorithm by clustering coefficient and common neighbor similarity addressed for large scale networks [69]. After that algorithm identify the initial communities, the expansion of the initial community is carried out by using a new weighted belonging degree. The selection of communities is based on the local clustering coefficient and the degree of similarity in common neighbors among their members. In this work, nodes are added to community if weighted belonging degree is greater than threshold. This evaluation concept is used for forming initial community and extending community.

Instead of taking node for seed, [70] designed an idea which take k clique as core community. The underlying assumption of the proposed algorithm is that cliques serve as the core of communities, since cliques provide a representation of the local characteristics of a community. The algorithm enhances the search efficiency by selecting a single node with maximum density as the initial community, thereby avoiding the creation of a large number of near-duplicate community structures. The k clique-based methodology meets NP hard problem. Moreover, it is inconvenient for sparse networks and possible for densely connected networks.

A. Choumane et.al [71] modeled a core expansion algorithm without computing modularity evaluation to detect communities. The whole procedure is completely independent from modularity and automatically detect the core of each possible community in the network. Then, iteratively expand each core by adding the nodes to form the final communities and used neighborhood overlapped measure in expansion process.

Compared to global methods, local community detection methods can identify communities within complex networks without relying on integral structural information. But, existing methods for overlapping community detection may have quality and stability deficiencies. For performing effectively and efficiently in complex networks, therefore, a new local community detection algorithm is modeled, called InfoNode [72]. This algorithm is a local community detection method that uses internal force between nodes. In that research, firstly, try to identify core members (seeds) of communities by using local degree central nodes and Jaccard index. Thus the method ensures that the selected seeds are central nodes within communities. Secondly, the fitness function extends the pre-selection of the node with the maximum degree among the seeds at each time. Finally, make a process to expand the top k nodes with the best performance from a pre-existing process, using a fitness function that considers the internal force between nodes. The objective of this method is to generate high-quality communities within the network.

The force that exists between two nodes is known as the internal force. The fitness function based on internal force is used to group nodes together based on their likelihood of forming a community. The more strongly connected nodes are with each other, the more likely they are to be grouped together into a community by the fitness function. Parameter ∞ is the same value of LFM and it is set to 0.8 to 1.2. However, that algorithm is for only disjoint communities.

Another local community detection method, the Label Propagation Algorithm (LPA) is one of the most widely used methods for detecting both disjointed and overlapped communities in a network. It is noted that the LPA algorithm is simple and fast and has three phases: initialization, propagation and filtering phase. In initialization, basic LPA, initially each node is assigned a unique label also called an identifier. This label identifies the community to which this node belongs. Then calculate the *node importance* of all vertices of the graph. If communities overlap, a node in the network can belong to more than one community. Therefore, to detect overlapping communities using LPA, the algorithm allows a node to contain multiple community identifiers. Specifically, a set of pairs (c, b) is associated with each node x, where c is a community identifier and b is the confidence coefficient of its direct neighbor. This coefficient b represents the degree of belonging of node x to community c and the importance of a node that sends this label. As propagation, to update node label in iteration *t*, this process is based on its neighbors' labels in iteration $(t-1)$ and the updated labels in the same iteration *t*. However, LPA has a major disadvantage in its instability, which is caused by the random update process used by the algorithm.

Therefore, a new algorithm, called Node Importance based Label Propagation Algorithm (NI-LPA) [73] was proposed to sort the nodes in a fixed order to solve the LPA instability problem, avoid the random selection of the node to be processed first and allow the algorithm to converge to a stable result. At propagation level of that algorithm, two cases are possible:

- In the first case: if multiple nodes have assigned the same community label c to the node v with different determined coefficients, the algorithm will sum up those coefficients to determine the overall degree of belonging of node v to community c.

- Second case: if the node v does not contain the label c among the set of labels already established, then, the couple (c, NI(c)) is added to its set.

After the propagation phase, each node contains a list of pairs (Label, Belonging coefficient) which represent the community labels assigned to the node and their corresponding degree of belonging coefficients. The algorithm filters out labels that are considered useless because their coefficients are very low compared to the other labels assigned to the same node. This filtering is done by deleting the pairs with belonging coefficients less than some threshold value. The specific threshold value chosen for this filtering process is 0.4, which is explained as the value below which a coefficient is considered of minor importance in this work.

For both weighted and unweighted graphs, K. Berahmand [74] proposed a community detection algorithm that involves detecting and expanding core nodes by considering their membership in a node and utilizing the strong community definition for graphs with weights. A subgraph C is considered a strong community if each node in C has more edges within C than edges with any other subgraph. Based on this definition, a node is said to belong to a strong community C if it has more edges with the subset of nodes in C (∞) than with nodes in other communities (β). If $\infty \geq \beta$, then $\infty \geq \infty + \beta \geq 0.5$, which indicates that a node is considered a member of a strong community only when the proportion of edges it shares within the community compared to all its edges (the total of edges it shares within the community and with nodes in other communities) is higher than 0.5.

For weighted graphs, ∞ represents the sum of a node's edge weights with other nodes within the community C, while represents the sum of a node's edge weights with nodes outside of community C. $\infty + \beta$ represents the total strength of the node. Based on this membership-degree function, a node is added in the initial subset if the ratio of the total weight of its edges with the initial subset compared to the total weight of all its edges exceeds γ. It can be seen that in detecting communities, the more accurate the weights of edges detected based on the node similarity, the more precise the local algorithm will be. Furthermore, the algorithm can identify the role of nodes within the network, such as core or outlier nodes, which can provide valuable insights into the structure and function of the network.

Most community detection algorithms face scalability challenges when applied to large, complex networks containing millions of nodes. To solve this issue, an effective method called the OCLN (local expansion-based Overlapping Community detection algorithm using Local Neighborhood information) has been proposed [75] in 2020. This method is different from most existing local-expansion methods and other methods (use all neighbors to get a community) like link community and label propagation in that it only considers key neighbors for community expansion. This reduces the number of nodes evaluated during expansion, making the method more efficient and scalable for large scale networks containing millions of nodes. OCLN is developed for overlapping-community detection in large-scale complex networks. In OCLN, the proposed expansion method iteratively expands a community until no neighbor can be added. A measure for evaluating the probability of a node belonging to a community (the belonging coefficient) is also proposed for removing misidentified nodes during the expansions. Instead of all nodes in the neighborhood, OCLN only uses some key neighbors in each expansion to achieve fast community expansion. This resolves the issue of the high cost of expanding process for existing local methods that are used to uncover overlapping communities.

Many community detection approaches face the difficulties such as there is no one-size-fits-all solution that can provide high-quality communities with high ground truth correspondence in reasonable execution time for all types of networks. In 2021, The GREESE [76] algorithm generates communities that have a high modularity and greater accuracy in less computational time. It proposed a coupled-seed expansion method for uncovering overlapped communities. This process consists of four phases: seeding, expanding, propagation and merging phase. Firstly, Specifically, construct a coupled-seed $s = \{v_i, v_j\}$by choosing a node and its most similar neighbor by $C(v_i, v_j) = |neighbors(v_i) \cap neighbors(v_j)|$. For expanding phase, this coupled-seed is expanded by using a fitness function that improves the quality of local communities' identification. In the propagation phase, if no coupled seed is generated due to two nodes have no common neighbors, assign each of these nodes to the elementary group that contains the highest number of their neighbors. As final phase, communities that share more than half of their members are joined together and then communities that share at least one third of the nodes in the smallest community are also merged. This work specified the threshold value of fitness from 1 to 0.5 for expansion process. This threshold determines the neighbors to belong to the local community.

Recently, L. Yan et al proposed an effective algorithm, named Two Expansions of Seeds (TES) [77]. TES used the topological feature of network nodes to find the local maximum nodes as the seeds which are based on the gravitational degree. The greater the gravitational degree of the node, the greater its influence and the stronger its information transmission ability in the network is, which makes the community discovery robust. After the seeds are identified using the gravitational degree, the algorithm uses a greedy strategy based on a fitness function to expand the communities. To improve the accuracy of community discovery, TES also employs a community cleaning strategy that avoid negative fitness. After that, the gravitational degree is implied again to expand the communities for a second time. This ensures that all nodes in the network belong to at least one community. Once the communities are formed, the next step is to merge similar communities. This is done by calculating the distance between communities, which is a measure of how similar or dissimilar the communities are. Communities that are too similar are merged together to obtain a less redundant community structure. If the distance between communities C1 and C2 is greater than threshold of the distance parameter, communities C1 and C2 are merged into one community since they overlap excessively.

In recent years, most researchers have applied on LFM’s fitness function to evaluate node’s quality in determining if a node should be belong to specific community However, the algorithms still face issues about instability of communities’ structure because of community’s size controlling parameter of LFM’s fitness evaluation function. Thus, researchers have designed models using their created various quality evaluation function with good seed selection techniques in this research area. But, although some algorithms proposed evaluation method to avoid parameter influences for adding nodes to the appropriate community, some can perform effectively on small networks and some can perform on only large scale network.

CHAPTER (3)
METHODOLOGIES FOR OVERLAPPING COMMUNITY DETECTION

This chapter introduces definition of community detection, their challenges and applied areas. As the background theory, local expansion strategy which is an overlapping community detection methodology is described with example calculation and the similarity formulas to find the similarity between pairs of nodes are also described.

3.1 Community

The most basic question for community discovery is “what is a community”. Generally, network communities are groups of nodes and these nodes within a group are more connected to each other than to the rest part of the network. In social networks, detecting communities is done by identify groups of nodes that are more similar to each other than to nodes outside the group. That is, it is expected that there will be more connections between nodes within a community than between nodes in different communities. When the nodes of a network can be arranged to form a group such that the nodes are said to have connected internally, then that network is said to have a community structure. Figure 3.1 illustrates a community structure. Community’s definition can be can be broadly classified into two categories based on the various perspectives of definition to a community: local and global.

3.1.1 Local Definition

Local definitions of a community are determined by the people who live in that community and their shared experiences, values, and traditions. Specifically, a group of nodes is referred to a community by providing some required properties of the group such as nodes’ similarity in terms of their attributes or behaviors. A community has also some constraints to the group such as such as the minimum number of connections that nodes within the group must have with each other, or the maximum number of connections they can have with nodes outside the group. Local definitions of a community can be further classified into two categories based on the links considered in the definition of community.

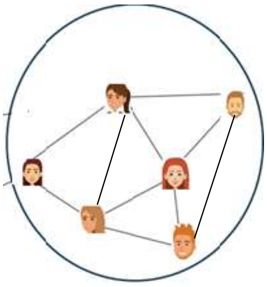


Figure 3.1 A community

The first category emphasizes the internal structure of a group of nodes, without considering their connections to the entire network. Generally, a community is a group of nodes that meets specific criteria and is not part of any other group that satisfies those same criteria. One way to define a community is by using the concept of a "clique", which is a subgraph in which every node is directly connected to every other node. Cliques have the highest possible link density of any subgraph, and so a community can be defined as a maximal clique. However, the requirement of a maximal clique is often too rigid to be a practical definition of a community, especially for larger networks. To combat this problem, Palla et al. suggested using clique percolation as a way to define communities. According to the n-clique-based definition of community, any two nodes within a community must be within a distance of no more than n, meaning there exists a path connecting them consisting of no more than n edges. However, this definition does not make any guarantees that a community' diameter has no more than n edges because there may be nodes within the community that are not directly connected by an edge, but can still reach each other through a longer path that goes outside the community.

The second category, strong and weak community definitions take into account both the internal edges of community and the edges between the community and the rest part of network. Strong community emphasizes the internal cohesion of the community, with each node being strongly connected to other nodes within the same community. In contrast, a weak community is characterized by the total degree of the nodes within the community being greater than the number of links that connect the community to the rest of the network.

3.1.2 Global Definition

Global definitions to community takes into account both the internal structure of a group of nodes and their connections to the rest of the network. One representative definition is in terms of network partition, which involves dividing the network into non-overlapping groups of nodes or communities based on the strength of their connections. To evaluate the quality of a network partition, various measurements are used. Once optimal partition of the network can be found based on one of these measurements each component of the partition corresponds to a community. The modularity definition, proposed by Girvan and Newman, is one of the most well-known global definitions of community. They used the configuration model as a null model to generate reference networks that have similar degree distributions as the real network, but where the connections are randomized. They then compared the modularity score of the real network partition to the average modularity score of many randomized networks. By using modularity, it is possible to detect the community structure by optimizing the modularity to identify the best possible partition. The proposal of modularity has been a significant driver of research in community detection and has greatly propelled the development of this field.

Reichardt and Bornholdt proposed an extended modularity based on the Potts model [78]. In addition, Rosvall and Bergstrom [79] proposed a different approach to community detection based on the concept of information theory. That approach uses the expected description length of a random walk on networks to identify community structure. Moreover, several probabilistic methods have been proposed to model the network data and detect communities.

The global definitions are accepted more widely than local because it lies in that Global definitions of community in complex networks focus on studying the network structure as a whole and aim to identify salient structural regularities that correspond to communities [80].

3.2 Community Detection

Community detection or identification of network structures is the process of partitioning a network or graph into clusters or communities where nodes within the same cluster are more closely interconnected with each other than they are with nodes in other clusters. A network or graph is a collection of nodes or vertices that are connected to each other by edges or links. Nodes represent individual entities in the network, while edges represent the relationships or connections between those entities. A graph can be either directed, meaning the edges have a direction or flow, or undirected, meaning the edges have no direction. The degree of a node in a graph is the number of edges connected to that node. Graph mining, also known as complex network analysis or network science, refers to the process of analyzing the structure and properties of a graph or network to extract useful information from it. This may involve exploring the graph to understand its topological properties, identifying important nodes or communities within the graph. Community detection or clustering is one subtopic within graph mining that involves identifying groups of nodes within the graph that are more closely connected to each other than to nodes in other groups.

Mathematically, when considering a graph G(V;E) where V represents the set of nodes and E represents the set of edges, community detection approaches aim to partition the node set V of a graph G(V;E) into k disjoint communities or clusters, denoted by C = {c1; c2; -----; ck}, where each node belongs to exactly one community. In other words, the communities are non-overlapping and exhaustive, meaning that every node in the graph belongs to exactly one community. The size of each community is denoted by Ni, and the total number of nodes in the graph is (v_i ϵ c_j , $N = \sum_{i=1}^{k} N_i$).

Figure 3.2 shows a basic visualization of the community structures present within a graph. At that figure, nodes and edges make up a graph, with the various colors representing different communities. It is clear from the visualization that there are five distinct communities, each with its own unique color. If examine each community, can observe that the nodes within each community are strongly interconnected, while there are only a small number of edges linking nodes across different communities. The goal of community detection is to partition the nodes in a graph into groups, or communities, such that the nodes within each community are more densely connected to each other than to nodes in other communities.

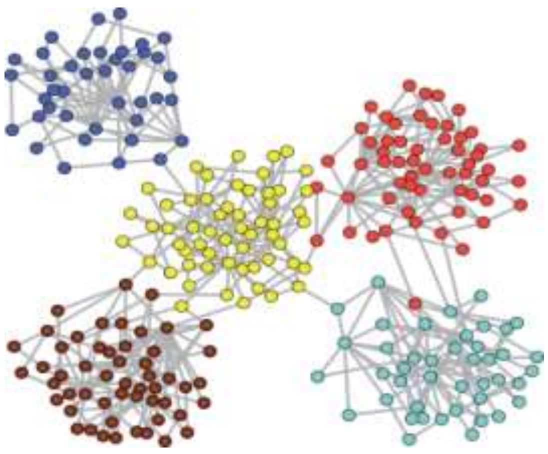


Figure 3.2 Community structure (same community with same color)

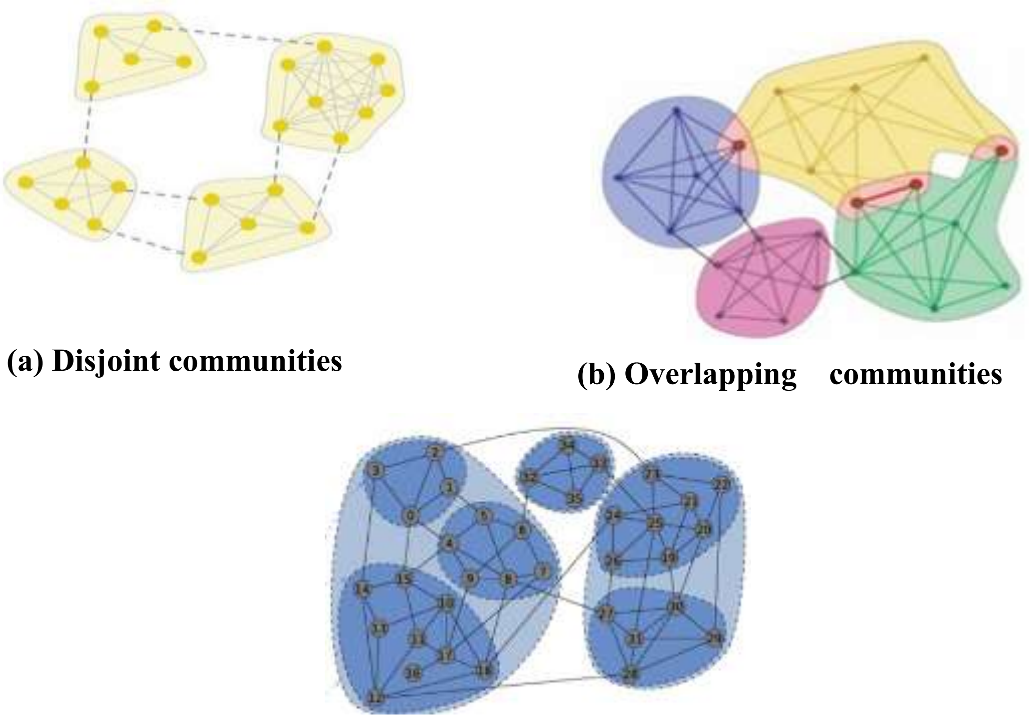
This can be achieved through either disjoint community detection, where each node belongs to exactly one community, or overlapping community detection, where a node can belong to multiple communities. Besides the basic definitions of concepts presented earlier, Table1.1 provides a comprehensive glossary of terminology for graph representation and community notations used in this dissertation.

Table 3.1 Glossary of terminology

Terms	**Definitions**
G(V,E)	Graph G with node set V and edge set E.
C	C = {c_1; c_2; -----; c_k} is the discovered community partition
c_j	the j^{th} community belong to C
N	Total number of node in graph
v_i	Each node within a community

3.3 Types of Community Structures

Community structures can be disjoint or overlapping, but overlapping community structures are often observed in real-life networks.



(a) Disjoint communities

(b) Overlapping communities

(c) Hierarchical communities

Figure 3.3 Three types of community structure

For example, a professor collaborates with researchers in different fields. Also one person can be a part of multiple groups at a time like group of family members, friends circle and clubs. In that cases, thus, nodes in a network have multiple roles or affiliations, and overlapping community detection can capture this complexity. Overlapping community detection is a challenging task, as it requires identifying groups of nodes that share multiple connections and have multiple roles or affiliations within the network. In addition to overlapping community structures, networks often exhibit a hierarchical organization, with communities embedded within other communities. These types of structures are shown in Figure 3.3.

3.4 Types of Graphs

Graphs can be of various types, each with its own characteristics and challenges. In recent years, with the increasing availability of data and advancements in graph mining techniques, researchers have been working with more complex graphs derived from real-world scenarios. Particularly, the following section presents five primary categories of graphs that are commonly encountered in research studies and have strong ties to real-world situations. These categories include heterogeneous and multi-layer graphs, sparse graphs, dynamic graphs, large graphs, and attribute graphs. Social networks are a type of attribute graph where nodes represent individuals and edges represent social connections. However, there are other types of networks that exhibit similar characteristics, such as collaboration networks, where nodes represent researchers and edges represent collaborations, or transportation networks, where nodes represent cities or transportation hubs and edges represent transportation routes.

3.4.1 Heterogeneous and Multi-layer Graph

The researches related to community detection in heterogeneous and multi-layer graphs [81] is summarized collectively. Because they both share the feature of having multiple types of nodes or edges, while heterogeneous graphs and multi-layer graphs may have different characteristics.

While a multi-layer graph is made up of several single-layer graphs, a single-layer graph consists of only one type of node/edge. Multi-layer graphs can capture different aspects of a complex system or phenomenon, such as different time periods, locations, or contexts. Detecting communities across multiple layers can reveal important patterns or trends that may not be apparent in individual layers. For example, in a social network, different layers may represent different types of interactions, such as friendship, collaboration, or communication. Community detection across these layers can help identify groups of people with similar interests or roles.

In heterogeneous graphs, different types of nodes or edges may represent different entities or relationships, and detecting communities across these different types of nodes or edges can provide valuable insights. The researchers have introduced a graph structure known as "metapath" [82]. This structure represents a path that connects different types of nodes with distinct edges. It can be considered a universal structure that represents the semantics of a path. For example, a metapath in a collaboration graph can be “Organization-(affiliated with) → Author-(written by) → Paper-(publish at) → Venue” and where it includes four node types (“Author”, “Venue” “Organization” and “Paper”) and authors can collaborate with each other. Thus, there are also self-relationship in author node and four types of edges (“affiliated with”, “written by”, “publish at” and “co-author”). In this study, specifically, metapath selection has been combined with user-guided clustering, which involves using prior information about a small set of nodes and their communities. Community detection in such a graph can help identify groups of authors working in similar areas or venues, or highlight important papers or venues in a field.

Figure 3.4 demonstrates the distinction between a heterogeneous graph and a multi-layer graph. In reality, a multi-layer graph can be viewed as a specific type of heterogeneous graph.

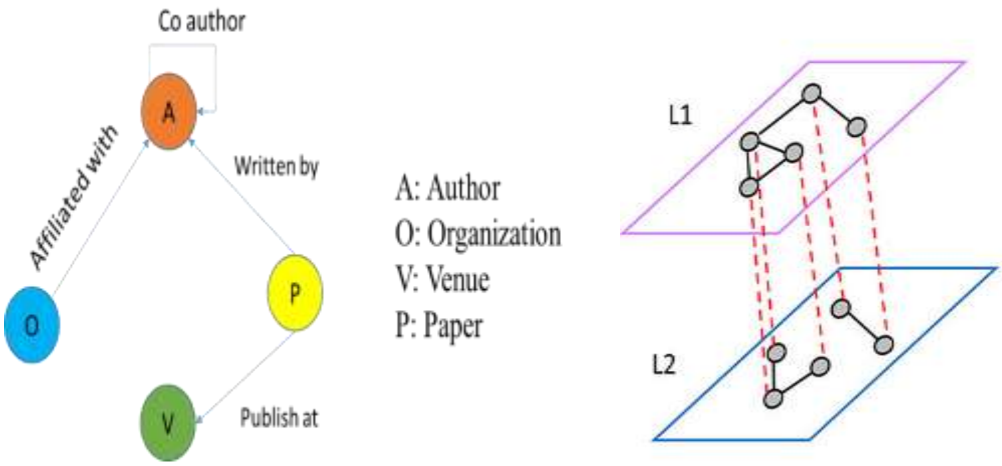


Figure 3.4(a) Heterogeneous graph (b) Multi-layer graph

3.4.2 Sparse Graph

Sparse graphs are frequently encountered in graph mining as opposed to dense graphs. But, the distinction between dense and sparse graphs is not always clear-cut and depends on the specific application or problem being considered. A graph is considered dense if the number of edges is $|E| = O(|V|^2)$ and the maximum number of edges in a directed graph is $|V| (|V| - 1)/2$, whereas a graph is sparse if it has a number of edges that is $|E| = O(|V|)$. Understanding sparse graph community detection is very important in many real-world scenarios because most large-scale networks, such as social networks, biological networks, and communication networks, are naturally sparse and incomplete. Traditional spectral clustering methods may not be the most effective approach. For this reason, one alternative approach that has been proposed is to use non-backtracking random walks on the graph to construct a refined matrix, which captures higher-order structural information and can help overcome the limitations of spectral clustering on Laplacian or original adjacency matrix.

The researcher, Atieh [83] proposed a novel approach for community detection in sparse graphs by combining a perturbation strategy and link prediction. The basic idea of their approach is to make dense graph from the original sparse graph by completing partial open triangles, which are triplets of nodes that are connected by only two edges instead of three. The completion of open triangles is based on link prediction, which predicts the existence of missing edges in the graph. Through the application of this enhancement method, it is possible to obtain improved community partitions from denser graphs.

3.4.3 Dynamic Graph

Dynamic graphs, also known as temporal graphs or time-varying graphs, are a type of graph that captures the evolution of networks over time. In dynamic graphs, the structure of the graph can change over time due to the addition or removal of nodes and edges, or changes in the properties of the nodes and edges themselves. One important consequence of this dynamic nature is that the communities or clusters of nodes within the graph can also change over time. This is known as community evolution, and it is a fundamental aspect of many real-world networks, such as social networks, communication networks, and transportation networks. Figure 3.5 illustrates a dynamic community detection framework.

The DYNMOGA (Dynamic Multi-Objective Genetic Algorithm) model [84] is a genetic algorithm-based approach for detecting communities in dynamic graphs. It uses a multi-objective optimization approach that considers two objectives: accuracy and smoothness in the temporal graph at each timestamp. Additionally, it attempts to minimize the cost of transformation between two community partition results at consecutive timestamps. Then [85] proposes a prior & posterior stochastic block model (PPSBM) and combines a static model for individual graph snapshots with a temporal model to track the evolution of community memberships over time. To track the evolution of community memberships over time, the PPSBM uses an extended Kalman filter (EKF) [86], which is a recursive Bayesian estimation technique that updates the probability distribution over community memberships as new graph snapshots become available. The EKF utilizes the parameters learned from the previous snapshot to optimize the current community memberships, while also updating the parameters based on the current snapshot.

Like PPSBM, the SBTM [87] model extends the original SBM by adding a temporal component, which models the evolution of community memberships over time. In the SBTM model, the evolution of the network is assumed to follow a hidden Markov model, where the state of the network at each time step is determined by the community memberships of the nodes at that time step, and the transition between states is influenced by the presence or absence of edges between nodes in adjacent time steps

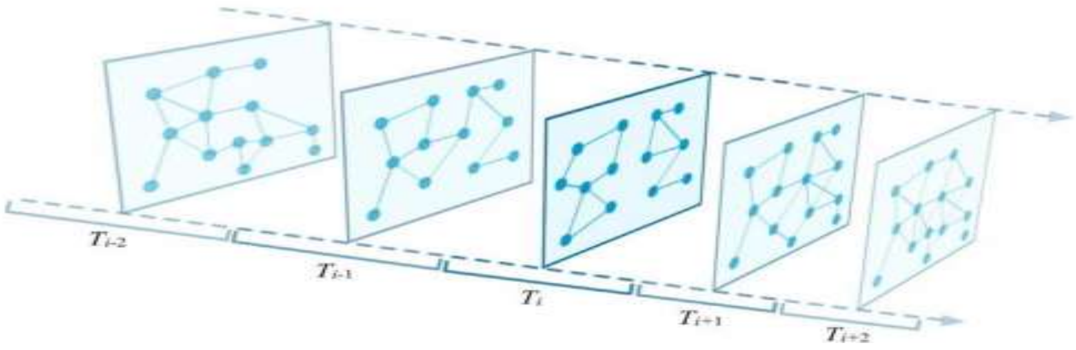


Figure 3.5 Temporal networks consisting of five time frames

3.4.4 Large Graph

Large-scale graphs are becoming increasingly common in various fields. As the size of these graphs can be in the millions or even billions of nodes and edges, traditional community detection algorithms may not be scalable enough to handle such large-scale data. In order to handle community detection in large-scale graphs, researchers have developed a range of algorithms and techniques that are designed to be more efficient and scalable.

A local searching model [88] appears to be a graph clustering algorithm that uses random walks to detect communities within a given graph. The algorithm is designed to run in approximately linear time. To deal with the challenge of working with large graphs, the authors of the paper propose a novel approach that involves sparsifying the graph and removing unimportant edges while preserving the main graph structure. This reduces the computational complexity of the algorithm while still maintaining the overall connectivity of the graph. To efficiently prune the graph by retaining only the top ranked edges for each node, the local sparsification algorithm is designed. The edges are ranked according to the Jaccard similarity between the current node and the other endpoint node. Additionally, a threshold is established to ensure that the pruned graph remains connected.

The GEM model [89] is a graph clustering algorithm that is designed specifically for large-scale social networks. The algorithm includes three main steps. The first step of the GEM model involves extracting the main skeleton from the original graph. In the second step, the GEM model applies a weighted k-means algorithm, along with an improved seeding strategy, to partition the filtered skeleton graph into communities. As the last step propagates the results of the skeleton graph clustering back to the entire graph. Specifically, the remaining nodes in the original graph are assigned to the communities generated in step two using a breadth first search (BFS) approach. This ensures that every node from the graph is allocated to a community, even if they were not included in the skeleton graph.

3.5 Challenges

There are various views of community in network analysis, each of which can be beneficial under different conditions. Here are some potential benefits of each view:

3.5.1 Cut Based View

Cut-based view is a way of defining a community in a network or graph by identifying groups of nodes with minimal connections to nodes outside the group. The internal structure of the group is not taken into account. This means that a group can be defined as a community even if it has weak internal connections or is not tightly interconnected [90]. This approach is utilized in the graph partitioning algorithm developed by Kernighan and Lin. To use the Kernighan-Lin algorithm, the user must specify the number and size of the groups they want to form. It is to find a partition that minimizes the cost function, which is defined as the difference between the total weight of edges within a group and the total weight of edges between groups. As Challenges of this view

- The internal connections within a group are not taken into account.
- There is no means of prioritizing groups with densely connected internal nodes group.
- The number of communities must be predetermined.

3.5.2 Clustering View

The goal of clustering is to maximize the internal similarity or density of objects within a cluster. The basic concept is to identify the groups of nodes in a network that are densely connected with each other and sparsely connected with nodes outside the group. Advantage of this approach is that the number of groups or clusters does not need to be predefined. The Newman-Girvan algorithm, also known as the Girvan-Newman algorithm, is a popular method for community detection or clustering in complex networks. It is based on the idea of edge and iteratively remove edges with high betweenness. The challenges are:

- A clear stopping criterion, such as Modularity, must be established.
- The utilization of Modularity parameters transforms the problem into an optimization problem.
- Determining an optimal clustering algorithm is challenging.

3.5.3 Stochastic Block Model View

Unlike the clustering view, the SBM does not aim to maximize internal density or minimize external links. Instead, it uses the concept of structural equivalence to identify groups of nodes that connect to nodes in other communities in a similar way. Structural equivalence is a notion from social network analysis that refers to the similarity of the roles or positions of nodes in a network. The advantage of the SBM is that it can handle a wide range of network structures, including bipartite graphs, directed networks, and weighted networks. It can also be used to generate benchmark datasets for evaluating community detection algorithms. However, the SBM requires more complex calculations and optimization methods than other approaches.

3.5.4 Dynamic Nature of Communities an Issue

Most of the work in community detection has been focused on static networks. However, many real-world networks, such as social media networks, are dynamic and constantly evolving, with nodes and edges appearing and disappearing over time. In dynamic networks, the communities or clusters of nodes can also change over time, and a dynamic community can be represented by an ordered pair of (nodes, periods), where nodes represent the clusters of nodes during any given time period, and periods represent the time duration. As dynamic communities evolve over time, different scenarios arise during the detection of these communities. These are described below and illustrated in the Figure 3.6.

- **Growth**: New nodes can be added to an existing community over time, leading to an increase in the size of the community.
- **Contraction**: Some nodes can leave a community over time, leading to a decrease in the size of the community.
- **Merging**: Different communities can merge over time, resulting in a larger community with a new set of nodes and edges.
- **Splitting**: One community can split into two or more communities over time, resulting in smaller communities.
- **Birth**: A new community can emerge that did not exist in earlier time intervals.
- **Death**: A community can completely disappear at any time.
- **Resurgence**: A community may remain dormant for a certain period and then reappear as if nothing happened.

Challenges encountered in the research of dynamic communities include:

1) Most of the existing community detection algorithms were designed for static networks and do not take into account the dynamic nature of communities.

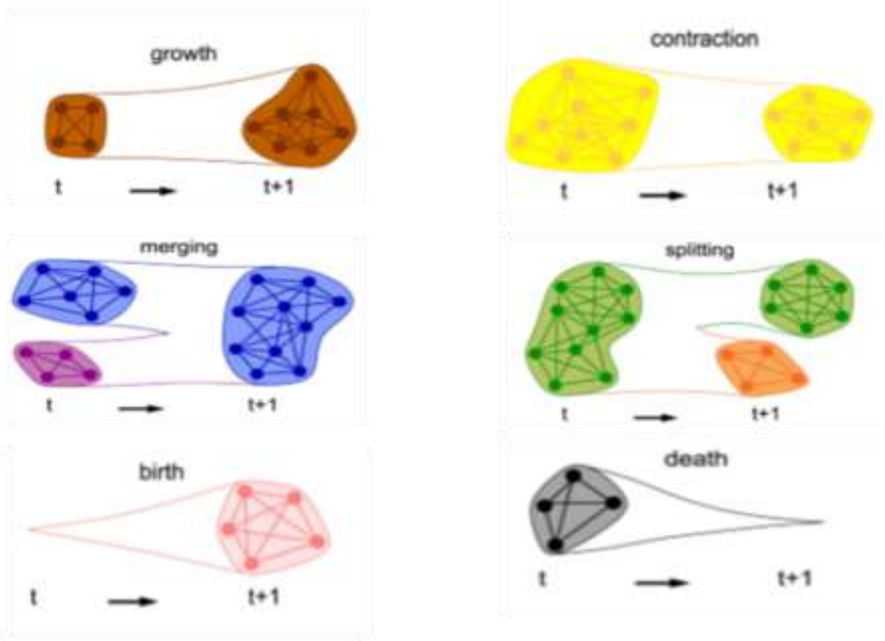


Figure 3.6 Communities evolution in dynamic networks

1) Defining dynamic communities is more complex than defining static communities because dynamic communities are not just nodes and edges that remain constant over time.
2) To obtain better results, multiple snapshots of the community must be considered and evaluated.

To address those challenges, one approach to dynamic community detection is to use traditional algorithms for static community detection in each graph, and then compare the communities across different time steps to identify changes in the community structure over time. In addition to challenges of the above view involved, overlapping community structure became an issue.

3.5.5 Communities Overlap an Issue

In the early days of community detection research, it was common to assume that communities in networks are disjoint. However, many real-world networks, including social networks, exhibit community structures that are not strictly disjoint. Instead, nodes can belong to multiple communities simultaneously, which is known as overlapping community structure. In real world, it's common for an individual to belong to more than one community. Therefore, the consideration of the "Overlap" characteristic of communities cannot be neglected.

3.6 Application Areas

Community detection is a cross-disciplinary field that has attracted researchers from various domains. As a result, there have been numerous studies on community detection, each with its own focus, assumptions, and limitations [91].

3.6.1 Ecommerce

E-commerce platforms enable users to buy and sell goods and services, as well as transfer funds and data over a network. Depending on the type of transaction, e-commerce can be divided into different categories such as B2B (business-to-business), B2C (business-to-consumer), and C2C (consumer-to-consumer). Social media platforms, on the other hand, enable users to connect with each other based on common interests and interact online. Users may form communities based on their interests and online behavior, creating a network of like-minded individuals. These communities can be leveraged for e-commerce purposes by detecting online groups and marketing products or services to them in a targeted and efficient manner.

3.6.2 Criminology

Community detection can be used to identify criminal user groups on social media platforms, including groups that are made up of both real person accounts and bot accounts. These groups may use social media to coordinate illegal activities, disseminate criminal or terrorist propaganda, and recruit new members. The authors [92] used community detection to identify communities within criminal networks, which can help law enforcement agencies to better understand the structure and dynamics of these networks. Pinheiro's study focused on identifying fraud events on telecommunication networks by using community detection to analyze customer behavior and detect outliers that may indicate fraudulent activity. This can help to prevent and investigate fraud in the telecommunications industry. Similarly, Waskiewicz's study used community detection to detect terrorist group activities on online social networks. By analyzing the connections between users and identifying communities that exhibit suspicious behavior, this approach can help to identify and prevent terrorist activities online [93].

3.6.3 Public Health

Community detection is used in the health domain for various purposes, including detecting disease outbreaks and identifying subgroups of patients with similar characteristics or conditions. Salathe and Jones studied the impact of community structure on the spread of infectious diseases. In cancer research, community detection has been used to identify different types of tumors and to group patients with similar tumor characteristics.

Bechtel et al. [94] proposed a community-based approach for lung cancer detection, which used community detection to identify groups of patients with similar gene expression patterns and clinical features. Haq and Wang also used community detection to analyze genomic datasets for 12 different types of cancer. By identifying subgroups of patients with similar genetic profiles, they were able to predict survival rates and identify the distribution of tumor types across these communities.

3.6.4 Politics

Community detection can be used in political science to understand the power relations between individual persons and political parties. With the growth of social media platforms like Twitter, Facebook, Instagram, etc., there is a lot of information available related to different political parties and their communities. Researchers can use community detection to understand the thinking and preferences of different political communities, and how these communities are connected to each other. This can be particularly useful for political parties to understand the public sentiment, and to craft their messages and strategies accordingly. However, fake news can also spread rapidly on social media, and political parties may use these platforms to spread misinformation. The political community plays a vital role to recommend and connect to the people of our country.

3.6.5 Smart Advertising and Targeted Marketing

Community detection can be used by companies to analyze customer behavior and group them into different segments based on their preferences, interests, and purchasing patterns. By doing so, companies can tailor their marketing strategies to specific customer groups. Additionally, community detection can help companies identify influencers within their customer base, who can then be targeted to spread the word about the company's products or services to their followers.

3.6.6 Recommendation Systems

By identifying communities of users who have similar preferences, recommendation systems can suggest items (e.g., books, movies, songs, products) that a user may be interested in. For example, if a user is part of a community of fans of a particular author, a recommendation system can suggest other books by the same author or books in the same genre. Similarly, if a user is part of a community of fans of a particular band, a recommendation system can suggest other songs or albums by the same band or by similar artists [95].

3.6.7 Social Network Analysis

Community detection is a useful tool for understanding the structure of complex networks, including social networks. Social Network Analysis (SNA) is a field that focuses on the study of social networks, their properties, and their impacts on individuals and society. Community detection is a key technique used in SNA to identify groups or clusters of nodes (individuals or entities) and identify subgroups within a larger community of users on social media platforms like Facebook, Twitter, or LinkedIn. These subgroups might be based on shared interests, demographic characteristics, or other factors, and can provide insights into how people interact online.

3.7 Local Community Detection

Local community detection algorithms aim to identify a set of nodes within a larger network that are densely connected and contain the given query nodes. It can be seen as a personalized task because the focus is on a specific subset of nodes rather than the entire network. Global community detection algorithms aim to partition the entire network into disjoint communities, which can be a time-consuming process, especially for large networks. Although there are numerous algorithms designed for detecting global communities in networks, most research on community detection has concentrated on local algorithms. In many real-world scenarios, the focus is on local communities rather than global ones. For example, social networking applications such as Facebook and WeChat suggest potential friends to a user by analyzing their local community. This is because individuals who belong to the same social circles as the user are more likely to have similar interests and preferences [9]. Therefore, local community detection algorithms are preferable to global community detection ones in such scenarios, as they can provide more targeted and personalized recommendations or interventions based on the properties of the local community.

3.7.1 Local Expansion Approach

Local expansion is another type of overlapping community detection algorithm which is based on nodes. The main idea of local expansion is to choose a seed community and then expand from it by using various fitness function. The following example explains local expansion approach by using fitness evaluation function based on inter degree and outer degree. In this strategy, the implementation results differ by relying on fitness function that researcher used.

Example

This example utilizes a random graph and describes the step by step process of local expansion approach on that random graph of Figure 3.7.

Step1: Select Seeds

For this step, choose the nodes {1}, {5}, and {8} randomly from the graph's nodes as the seed set. These nodes are shown in grey color for illustration.

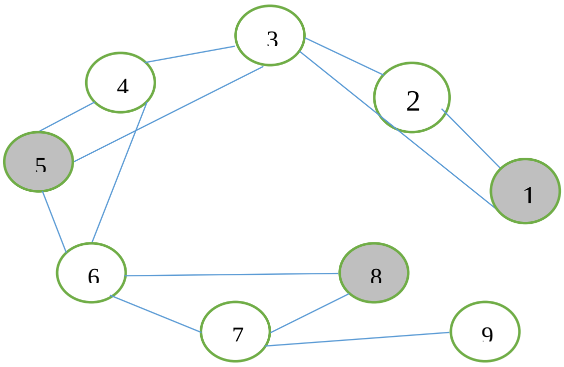


Figure 3.7 Example random graph

Step2: Expand the seed nodes

In this step, each seed is expanded based on fitness function. Before expansion, suggest distinct local communities for each seed node. Thus, this example locates three local community structure for each seed.

Expansion of seed {5}

Node 5 is expanded by maximizing a local fitness function, $w_{in}/ (w_{in} + w_{out})$, through the addition or removal of members. In applied fitness function, w_{in} is edges among members and $w_{in} + w_{out}$ is edges among the union of members and the neighbors. Members mean vertices selected to be members of a community. The neighbors refer to the adjacent vertices of the seed. In this case, 3, 4 and 6 are adjacent vertices of 5. Therefore, local structure of the following Figure 3.8 will be evaluated by using local fitness function to decide whether possible to add or remove members to explore a better community.

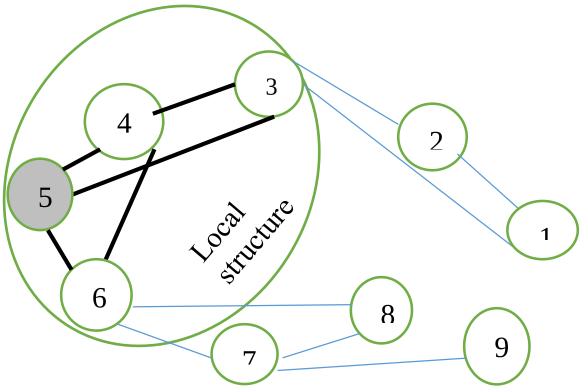


Figure 3.8 Local structure including seed {5}

At the beginning, a community have only one member in Figure 3.9 and there are no edges among the members, w_{in}=0 because the initiating member of local structure is node 5. The black edges contribute to w_{out}, w_{out}= 5. Fitness is 0/5=0.

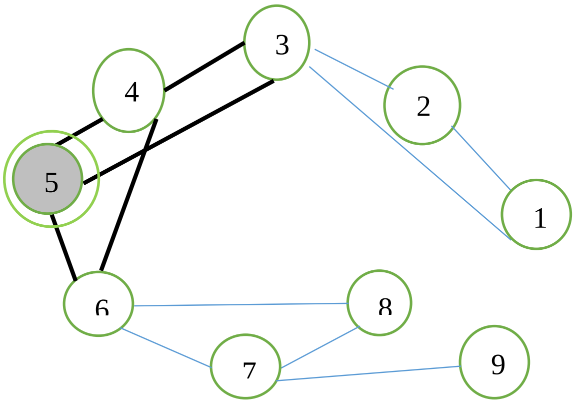


Figure 3.9 A community with a member

Upon the addition of vertex 4 to the local structure, the value of w_{in} becomes 1 (as there is only one edge between vertex 5 and 4 within the local structure), while w_{out}= 5. The resulting fitness value is calculated as 1/5, which is equal to 0.2. As a result, the fitness value improves, and Figure 3.10 is shown by adding vertex 4 to the structure with certainty. The neighbors of the local structure are 3 and 6.

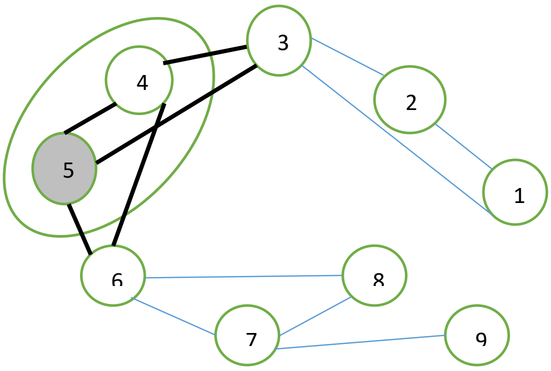


Figure 3.10 A community with two members

If vertex 3 is added to the local structure, and its neighbors are {6, 2, 1}, then the value of w_{in} = 3 and w_{out} is 5. As a result, the resulting fitness value is 3/8, which is equal to 0.4. Since the fitness value increases, vertex 3 is added to the structure. Therefore, the members of this structure are {3, 4, 5} in Figure 3.11.

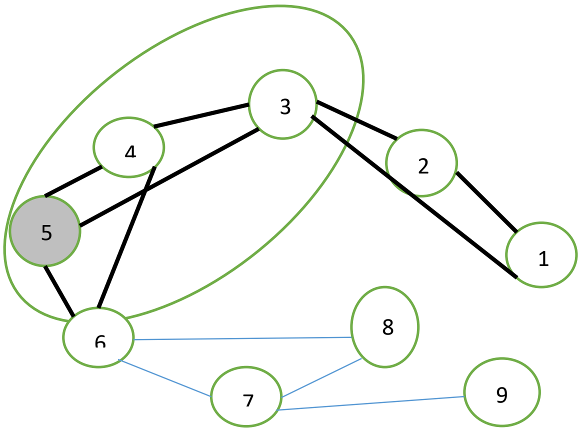


Figure 3.11 A community with three members

Subsequently, if vertex 6 is added to the structure and its neighbors are 1, 2, 7, and 8, the value of w_{in} becomes 5, while w_{out} is 6. Thus, the resulting fitness value is calculated as 5/11, which is equal to 0.45. Consequently, the members of the community structure are {3, 4, 5, 6}. It is shown in Figure 3.12.

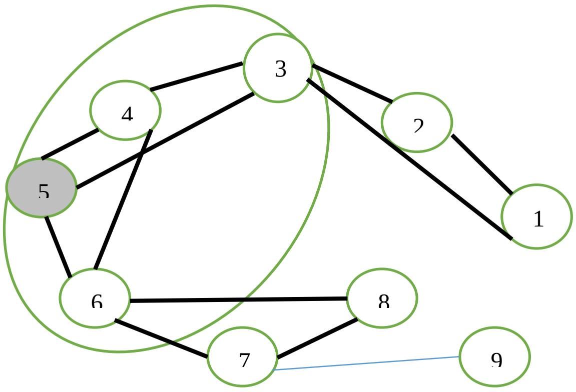


Figure 3.12 A community with four members

When vertex 2 is added, the neighbors of local structure are 1, 7, 8. In that case, w_{in}= 6, w_{out}=5, fitness is 6/11=0.55. Thus, vertex 2 is surely added to the structure in Figure 3.13 and the members are {2, 3, 4, 5, 6}.

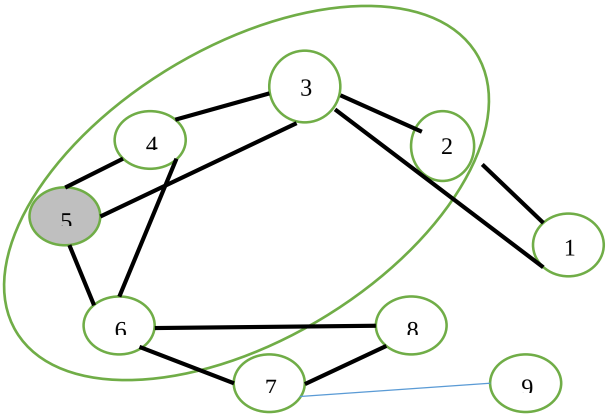


Figure 3.13 A community with five members

For vertex 1, w_{in}=8, w_{out}=3, fitness=8/11=0.7. Therefore, 1 is inserted to the community structure and members are {1, 2, 3, 4, 5, 6}. The neighbors are 7, 8 and all links related to the local structure are shown in Figure 3.14 with thick black lines.

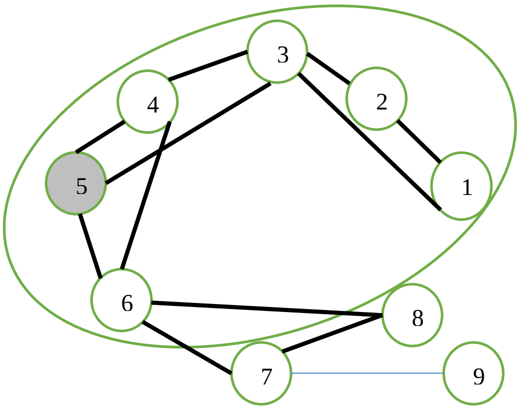


Figure 3.14 A community with six members

This strategy checks whether fitness value decreased or increased if withdrawal nodes from current local community structure. If removing node cannot decrease the quality value, that node should be removed from the community. Otherwise, that node is retained in the community.

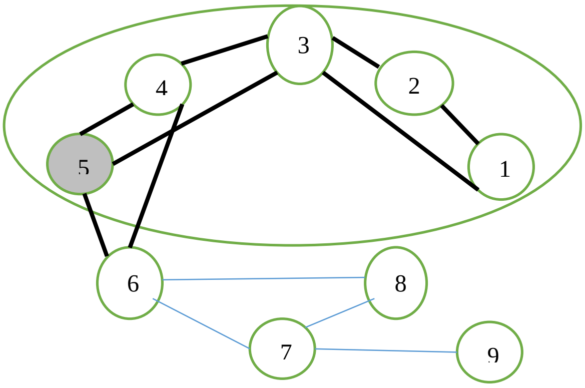


Figure 3.15 A community with remaining members after remove vertex 6

When try to remove vertex 6 from structure, local fitness is improved because w_{in}=6, w_{out}=2, fitness= 6/8=0.75. Consequently, removing 6 is not decreased the fitness, thus vertex 6 is taken out from local structure according to Figure 3.15. Then any members from the structure are not removed because their fitness decrease if remove them. This stage reaches the end of the seed expansion for the seed {5}. Therefore, the members of local community structure for seed {5} are {1, 2, 3, 4, 5}.

Seed Expansion result for seed {5}

By following the above procedure, the final local community of seed {5} is {1, 2, 3, 4, and 5}.

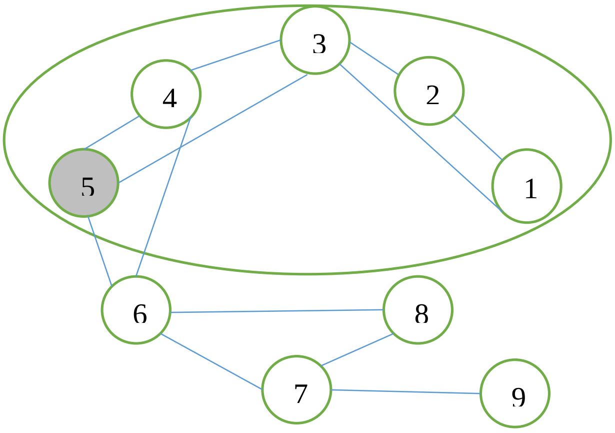


Figure 3.16 A local community after seed {5} expansion process

The remaining next seeds {1} and {8} are also extended in the same way as the expansion process of seed node 5.

Seed Expansion result for seed {1}

The final local community of seed {1} is {1, 2, 3} and it can be seen in the Figure 3.17 by following with above procedure.

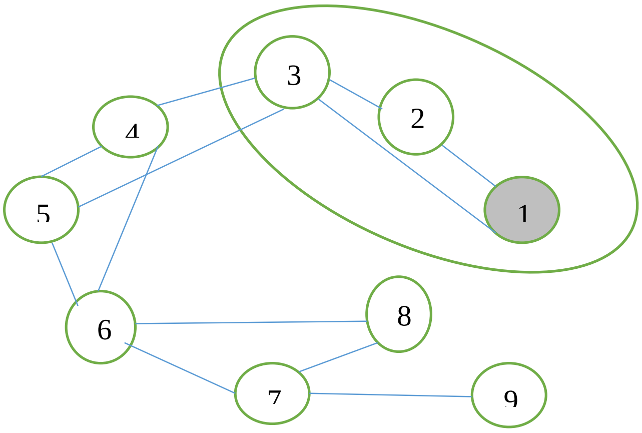


Figure 3.17 A local community after seed {1} expansion process

Seed Expansion result for seed {8}

The local community structure for seed {8} is {4, 5, 6, 7, 8, 9} and it is shown in Figure 3.18.

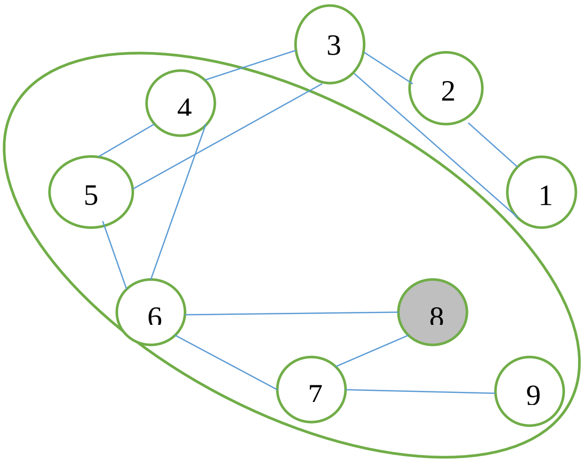


Figure 3.18 A local community after seed {8} expansion process

Step3: Merge Intermediate Communities

This step merges the all local community structures for seed {5}, {1} and {8}. When merged that three local structures, the overlapped communities and overlapped nodes are occurred in the following figure. With three overlapped communities, the overlapping nodes 1, 2, 3, 4, 5 are identified.

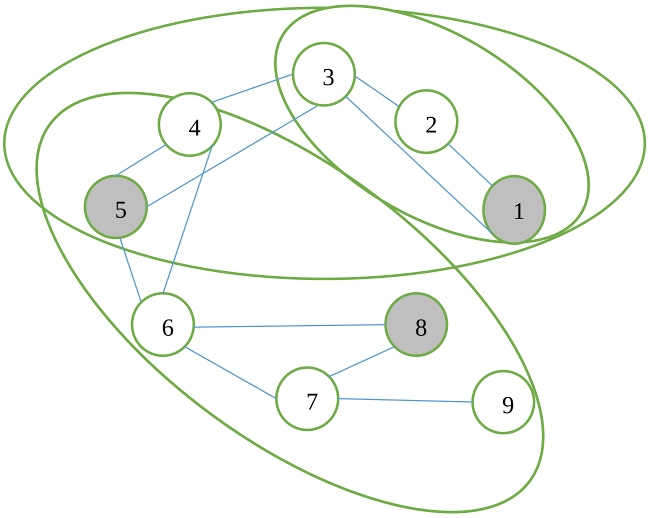


Figure 3.19 Overlapping communities with three local communities

3.7.2 Fitness Functions

In local community detection, the local communities are discovered by fitness quality optimization as well as modularity based optimization for extending seeds. Typical local modularities, R and M, were proposed by Clauset [96], Luo et al. [97] respectively and popular fitness function is proposed by Lancichinetti [13].

M function

Lou et al. proposed a local modularity metric M for detecting communities in complex networks. The local modularity metric M is defined as follows:

$$\mathrm{M} = \frac{M_{in}}{M_{out}} = \frac{\frac{1}{2}\sum_{ij} A_{ij}\theta(i,j)}{\sum_{ij} A_{ij}\lambda(i,j)} \qquad (3.1)$$

M_{in} represents the number of edges within a community C that have both endpoints in that community and $\mathrm{M}_{\mathrm{out}}$ represents the number of edges that have one endpoint in community C and the other endpoint outside of community C. If both nodes i and j belong to community *C*, θ (i, j) = 1, otherwise, θ (i, j) = 0. A (i,j) = 1 when just only one of nodes i and j belongs to community C and else λ(i, j) = 0. A high value of M indicates a strong community structure in the network, where the communities have more internal edges than expected by chance, and fewer external edges than expected by chance.

R function

The local community detection algorithm proposed by Clauset, which is based on the local modularity R. This algorithm defines a local community as a subset of nodes that form a subgraph, where the nodes on the boundary have more connections with nodes within the subgraph than with nodes outside of it. The local modularity R is defined as:

$$R = \frac{B_{in}}{B_{in}+B_{out}} \tag{3.2}$$

Bin denotes the number of edges whose endpoints are in the boundary part of the local community, and *Bout* denotes the number of edges connecting the boundary nodes with the nodes outside the community. However, the algorithm needs to define the size of the community, in advance.

Clauset's local community detection algorithm and the LWP method both concentrate on a specific portion of the subgraph to detect local communities. Equation (3.2) emphasizes only on the boundary nodes of the local communities, whereas equation (3.1) focuses only on the community center. As a result, both algorithms can only detect a portion of the actual local community.

F function

Lancichinetti et.al proposed a local modularity based algorithm for community detection that maximizes the fitness function f_G [13]:

$$f_G = \frac{k_{in}^G}{(k_{in}^G+k_{out}^G)^{\propto}} \tag{3.3}$$

Where $k_{in}^G\ and\ k_{out}^G$ represent the sum of the weights of the edges that are internal to the community G and external to the community G, respectively. The parameter α controls the size of the communities and is usually set to a value between 0 and 1. The fitness of a node i with respect to a community G is defined as the change in the fitness of the community G when node i is added to or removed from the community, i.e. $f_G^i =$

$f_{G+i} - f_{G-i}$. In this formula, if adding node i to community G increases the fitness of the community, then the fitness f_{G+i} is positive and node i is included in the community. If removing node i from community G increases the fitness of the community, then f_{G-i} is negative and node i is removed from the community. Where f_G^i> 0, means the value of fitness function increased with the node i joining community G, and thus node i should be included in community G; conversely, where $fG\ i$ < 0, means the point i should be removed from G.

3.8 Similarity and Distance Measures

Similarity or distance measures are core components used by distance-based clustering algorithms to cluster similar data points into the same clusters, while dissimilar or distant data points are placed into different clusters [98]. Generally speaking, the purpose of cluster analysis is to organize data into different groups: data in the same group are highly similar while those from different groups are dissimilarity. In graph clustering, popular Jaccard and Consine similarity measure to compute the similarity between communities or clusters among the following measurements.

3.8.1 Jaccard Similarity

The Jaccard coefficient is a metric that quantifies the similarity between two sets and is commonly used in graph analysis to measure the similarity of the neighborhoods of two vertices. It is computed as the size of the intersection of the neighborhoods of U and V divided by the size of the union of the neighborhoods of U and V. The Jaccard coefficient ranges from 0 to 1, where 0 indicates no similarity and 1 indicates complete similarity between the neighborhoods of the two vertices.

The Jaccard coefficient is also sometimes referred to as the Tanimoto coefficient. In the context of text documents, the Jaccard coefficient can be utilized to evaluate the similarity of two documents based on the presence or absence of certain words. It is computed as the size of the intersection of the sets of words present in both documents divided by the size of the union of the sets of words present in either document. Similarly, in graph clustering, the Jaccard coefficient can be applied to evaluate the similarity between clusters or nodes. The Jaccard Coefficient’s formal definition when it is expressed over a bit vector can be given by

$$J(u,v)=\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{3.4}$$

The Jaccard similarity is defined as the size of the intersection of the neighborhoods of two vertices divided by the size of the union of their neighborhoods. N (u), N (v) represents neighbors of vertex u and v, respectively. $N(u) \cap N(v)$ refers to the set of nodes that are neighbors of both u and v. $N(u) \cup N(v)$ represents the total number of neighboring nodes that have at least one adjacent vertex with both nodes u and v. A larger value of the Jaccard similarity indicates a higher degree of similarity between the nodes, and a smaller value indicates a lower degree of similarity. Figure 3.7 shows the example graph to compute similarity.

When compute the Jaccard for vertex A and vertex D, the intersection of their neighborhoods is the single node, vertex B. The size of the union is two, nodes C and B. Note that despite that both A and D are neighbors of B, count only B as one node in the union. This makes the Jaccard value 1/2.

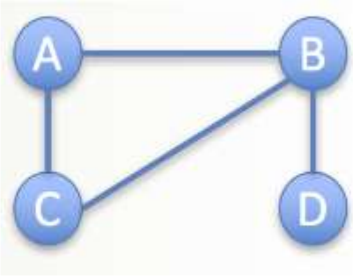


Figure 3.20 Small graph with four vertices

3.8.2 Cosine Similarity

Cosine similarity is a widely used method for measuring similarity between two term vectors, which are commonly used to represent documents in text mining. It calculates the cosine of the angle between two vectors in a high-dimensional space. The cosine similarity score between two document term vectors ranges from 0 to 1, where 0 indicates that the two vectors are completely dissimilar, and 1 indicates that they are identical. A higher cosine similarity score between two document term vectors indicates that they have more words or terms in common.

3.8.3 Euclidean Distance

Euclidean distance is a distance metric that measures the straight-line distance between two points in Euclidean space. In the context of text clustering, Euclidean distance is commonly used to measure the distance between two document term vectors. The Euclidean distance measure is a special case of Minkowski distance measure. Summation of all such squared lengths are taken, next, square root of the same is computed.

3.8.4 Minkowski Distance

Minkowski distance is a generalization of both Euclidean distance and Manhattan distance. Minkowski distance can perform well when all the datasets are compact and isolated. However, the statement that large-set attributes will dominate the others if the dataset is not able to fulfil this condition is not necessarily true.

3.8.5 Manhattan Distance

The Manhattan distance is a measure of distance between two points in a grid-like layout, such as a chessboard, where the distance is calculated as the sum of the absolute differences of their coordinates. It is called the Taxicab distance or City Block distance. Manhattan distance then refers to the distance between two vectors if they could only move right angles. There is no diagonal movement involved in calculating the distance. The Manhattan distance is a useful metric for measuring distance between points in a grid-like layout. But it may be less intuitive than the Euclidean distance in high-dimensional spaces.

3.9 Evaluation metrics

Evaluating the quality of a detected cover is a vital task, and there are various metrics available to measure the similarity of partitions, which can be extended to covers as well. Performance metrics are used to evaluate the quality of a detected cover by comparing it with a reference cover. Performance Metrics are metrics which measure the similarity of a detected cover with a reference cover. These evaluations can be measured by ground truth and no ground truth for both overlapped and disjoint communities. However, researchers extended some traditional evaluation metrics to measure quality of overlapped communities. That metrics are explained in the following and that are used in these experiments.

3.9.1 Performance Evaluation without Ground Truth

When measuring performance without ground truth information, modularity (Q) is popular evaluation metric. Naturally, when decomposes a graph, the nodes within a group densely connected with each other and the nodes between the groups sparsely connected. Therefore the connectivity within communities are measured by intra density and contraction. The symbol descriptions of evaluation metrics are shown in table 3.2.

Overlapped modularity (Q_{ov}): The modularity of overlap is an extension of the classical modularity and the edge-based extension of modularity is given by:

$$Q_{ov} = \frac{1}{2m}\sum_{c}\sum_{i,j\in c}[A_{ij} - \frac{k_i k_j}{2m}]\frac{1}{O_i O_j} \quad (3.5)$$

A higher Q_{ov} reflects a good community detection result.

IntraDensity: It measures connectivity within a community. Meanwhile, it represents density of community and how much densely connected nodes within a community. If high intradensity, the result is better.

$$IntraDensity = \frac{2*|E_c^{in}|}{|c|*(|c|-1)} \quad (3.6)$$

Contraction: It measures the average number of edges per node inside the community *c*. The larger the value of Contraction is, the better the community quality is.

$$Contraction = \frac{2*|E_c^{in}|}{|c|} \quad (3.7)$$

Table 3.2 Symbol descriptions

Symbol	Description
m	the total number of edges in the network
O_i , O_j	the number of communities of the vertex i and j belong to respectively

k_i, k_j	the degree of i and j respectively
A_{ij}	the element of adjacency matrix of the network.
$\|E_c^{in}\|$	the total number of edges in Community *c*
$\|E_c^{out}\|$	the total number of edges on the boundary of Community *c*

3.9.2 Performance Evaluation with Ground Truth

The Normalized Mutual Information (NMI) and Omega index are widely used in evaluation metrics with known data to measure good partition.

Normalize Mutual Information: The *NMI* was first introduced by Fred et al. as a measure of clustering quality and later extended to covers by Lancichinetti et al. [13]. The NMI measures the mutual information between the detected cover and the reference cover and normalizes it by the average entropy of the two covers. The resulting value is bounded between 0 and 1, where 0 indicates no similarity, and 1 indicates perfect similarity. Lancichinetti defined as follows:

$$ENMI(X|Y) = 1 - \frac{1}{2}\left[\frac{\mathrm{H}(X|Y)}{H(X)} + \frac{\mathrm{H(Y|X)}}{H(Y)}\right] \tag{3.8}$$

H (X|Y) denotes the normalized conditional entropy for cluster X with respect to cluster Y. The variables X and Y are stochastic variables that are associated with the partitions C and C', respectively. H (X|Y) is

$$\mathrm{H\ (X|Y) = H(X,\ Y)\text{-}H(Y)} \tag{3.9}$$

$$\mathrm{H(X,\ Y)} = \sum_{i\in X}\sum_{j\in Y} log\frac{X_i \cap Y_j}{N} + log\frac{X_i-(X_i\cap Y_j)}{N} + log\frac{Y_j-(X_i\cap Y_j)}{N} + log\frac{N-(X_i\cup Y_j)}{N} \tag{3.10}$$

$$\mathrm{H\ (X)} = \sum_{i\in X} log\frac{X_i}{N} + \log(1-\frac{X_i}{N}) \tag{3.11}$$

$$\mathrm{H\ (Y)} = \sum_{i\in Y} log\frac{Y_i}{N} + \log(1-\frac{Y_i}{N}) \tag{3.12}$$

Omega: The Omega Index [99] is a similarity measure that was introduced by Collins et al. as a way to evaluate the quality of overlapping community detection methods. It extends the Adjusted Rand Index (ARI) to handle overlapping communities. The Omega Index is based on pairs of nodes that are clustered in the exact same number of communities in both covers. Let *K*1 and *K*2 be the number of communities in covers *C*1 and *C*2, respectively. It defines as

$$\omega(C1, C2) = \frac{\omega_u(C1,C2) - \omega_e(C1,C2)}{1 - \omega_e(C1,C2)} \tag{3.13}$$

The unadjusted Omega index ω_u is defined as

$$\omega_u(C1, C2) = \frac{1}{M} \sum_{j=0}^{\max(K1,K2)} |t_j(C1)| \cap |t_j(C2)| \tag{3.14}$$

M equals to n (n− 1)/2 represents the number of pairs of node, and t_j (*C*) is the set of pairs appear exactly *j* times in a cover *C*.

The expected Omega index in the null model ω_e is given by

$$\omega_e(C1, C2) = \frac{1}{M^2} \sum_{j=0}^{\max(K1,K2)} |t_j(C1)| \cap |t_j(C2)| \tag{3.15}$$

The value of the Omega Index is highest at 1, which indicates perfect matching of the two covers. When there is no overlap, the Omega index reduces to the ARI.

F1-measure: This system is implemented by F1 measure at community level and was used in [100]. The F1 score is a common metric used to evaluate the quality of clustering or classification algorithms, and it combines both precision and recall into a single value. For each community identified by the algorithm, finds the best-matching reference community (i.e., the reference community that has the highest F1 score with the identified community). Precision is the fraction of the detected covers that match known complexes and it is defined as:

$$\text{P (C, C')} = \frac{(C \cap C')^2 / (C * C')}{|C|} \tag{3.16}$$

Recall represents the fraction of known complexes that match detected covers and it is denoted as:

$$\text{R (C, C')} = \frac{(C \cap C')^2 / (C * C')}{|C'|} \tag{3.17}$$

C means local community that given by the algorithm and C' indicates the actual community. F1 is defined as:

$$F\_score = \frac{2 * precison * recall}{precison + recall} \tag{3.18}$$

3.10 Chapter Summary

This chapter discusses basic concepts of the community detection, types of community structure, their challenges and application areas. The categories of overlapping community detection algorithms have already been described in the literature review of chapter 2. Therefore this chapter discusses an overlapping community detection approach, local expansion. It is also explained with the example calculation by applying random graph. Moreover, the similarity evaluation methods to find similarity between pairs of nodes are discussed and measurements to evaluate quality of overlapped community structure are also described.

CHAPTER (4)
SYSTEM ARCHITECTURE OF THE PROPOSED METHOD

This chapter describes the proposed system architecture, methodologies and algorithms for detection the overlapping communities. There are three phases in the proposed system design: seed identification, community expansion for local community detection and overlapped node identification. For first phase, seed identification algorithm is used and proposed community expansion algorithm is applied in second phase. Finally, overlapping vertices are detected by merging the derived local communities.

4.1 Methodologies

The graphs can be used to model many types of relations and process dynamics in computer science, physical, biological and social systems. A graph is a structure that comprises a set of vertices and a set of edges. So a graph is modelled to define the elements of two sets: vertices and edges as G (V, E) and social network is represented as graph model. Consider an undirected and unweighted graph, denoted as G (V, E). Here, V represents the set of vertices in G, and E represents the set of edges. Let C = {C1, C2, ..., Cn} denote the network community structure, which is a collection of n subgraphs where $C_i \in C$, C_i is the set of V. The notation descriptions of this chapter are listed in Table 4.1.

4.1.1 Extended Jaccard Similarity

In section 8 of chapter 3, traditional jaccard similarity equation (3.4) is described. This similarity is applied in finding similarity between pair of nodes to calculate the weight of each node for identifying seed.

In traditional jaccard index, their jaccard similarity index will be 1 if two datasets share the exact same members. Conversely, if they have no members in common, their similarity will be 0. Therefore, if there are no shared neighbors between pairs of vertices, the value of similarity can be zero. However, the seed node serves as the central node within a community, and its selection depends on the node's strength (i.e., the seed is a core node that is closely connected to many other nodes). Referring

Table 4.1 Notation descriptions

Notation	Description
$G(V,E)$	Graph consisting vertices and edges
$N(u) \cap N(v)$	Common neighbour of vertex u and v
$N(u) \cup N(v)$	Total number of neighbour of vertex u and v
k_{in}^{C}	Number of internal links within community
$k_{in}^{C} + k_{out}^{C}$	Total number of internal and external links of community
α	Community's resolution controlling parameter
ρ	Density of graph
k_{ext}	Number of external links of initial community
k_{int}	Number of internal links within initial community
d_{avg}^{G}	Average degree of graph

to equation (3.4), if there are no common neighbors between two vertices, a similarity value of zero is assigned, even if some nodes have a certain number of connections. Therefore, extended Jaccard Similarity is defined to avoid zero similarity by adding one in numerator.

$$Sim(u,v) = \frac{|N(u) \cap N(v)| + 1}{|N(u) \cup N(v)|} \tag{4.1}$$

The example calculation of similarity between pairs of vertices is explained in section 4.1.4. This evaluation is computed on pair of neighbor nodes of input graph.

4.1.2 Weight Evaluation

This weighted evaluation is grounded on the extended Jaccard similarity, which is defined as follows:

$$W_u = \sum_{u \in V} Sim(u,v) \tag{4.2}$$

This evaluation is conducted by summing up the similarities between each node and its neighbors. This approach enables the assignment of a core node or seed through weighted evaluation. Typically, vertices with a high degree in the network do not necessarily have a high weight. Therefore, the weight of a node in this study is not determined by the degree of each vertex. Instead, it is determined using equation 4.2. This weight is applied to identify core node among the nodes from the whole network.

4.1.3 Fitness Function

A community is a subgraph identified by the maximization of a property or fitness of its nodes. Lancichinetti et.al [13] proposed a fitness evaluation function f to measure tightly connected to the internal nodes of a community. This function is identified as follows:

$$f_C = \frac{k_{in}^C}{(k_{in}^C + k_{out}^C)^\alpha} \tag{4.3}$$

α is a positive real value parameter which is used to adjust the community scale. This quality evaluation function effectively measures the densely connected nodes within communities. Lancichinetti suggested this concept for each node (i) in the community, the number of internal links is more than the external links. That is, leads to the strong community and strong links with each other within a community if $k_{in}^C > k_{out}^C$. Otherwise, the community means weak community. On this idea, Lancichinetti developed equation (4.3) and conducted experiments in his research using a range of [0.6, 1.6], with the default value assumed to be 1. Consequently, the outcomes of different implementations are highly dependent on the parameter, and the best result is obtained by trying different parameter values.

4.1.3.1 Optimized Resolution Controlling Formula

In equation (4.3), the parameter variation leads to different outcomes in various implementations. As a result, the size of the community becomes unstable, and it is uncertain when the best result will be achieved precisely. To control the community scale in this work, parameter evaluation formula is defined as follows:

$$\alpha = \frac{log(k_{int} - d_{avg}^G) + \rho}{log(k_{ext})} \tag{4.4}$$

This formula is considered on the internal and external degree of initial cluster with density and average degree of the whole graph.

4.1.4 Example Similarity Calculation of Karate Social Network

To clarify the similarity between nodes of karate network, the following calculation is performed by using Jaccard index. This network has 34 nodes and the similarity for every pair which has link are occurred.

Sim(1,2) = 8/(25-7) = 0.4444444444444444

Sim(1,3) = 6/(26-5) = 0.2857142857142857

Sim(1,4) = 6/(22-5) = 0.35294117647058826

Sim(1,5) =3/(19-2) = 0.17647058823529413

Sim(1,6) = 3/(20-2) = 0.16666666666666666

Sim(1,7) = 3/(20-2) = 0.16666666666666666

Sim(1,8) = 4/(20-3) = 0.23529411764705882

Sim(1,9) = 2/(21-1) = 0.1

Sim(1,11) = 3/(19-2) = 0.17647058823529413

Sim(1,12) = 1/(17-0) = 0.058823529411764705

Sim(1,13) = 2/(18-1) = 0.11764705882352941

Sim(1,14) = 4/(21-3) = 0.2222222222222222

Sim(1,18) = 2/(18-1) = 0.11764705882352941

Sim(1,20) = 2/(19-1) = 0.1111111111111111

Sim(1,22) =2/(18-1) = 0.11764705882352941

Sim(1,32) = 1/(22-0) = 0.045454545454545456

Sim(2,1) = 8/(25-7) = 0.4444444444444444

Sim(2,3) = 5/(19-4) = 0.3333333333333333

Sim(2,4) = 5/(15-4) = 0.45454545454545453

Sim(2,8) = 4/(13-3) =   0.4

Sim(2,14) = 4/(14-3) = 0.36363363636366365

Sim(2,18) = 2/(11-1) = 0.2

Sim(2,20) = 2/(12-1) =   0.18181818181818182

Sim(2,22) = 2/(11-1) = 0.2

Sim(2,31) = 1/(13-0) = 0.07692307692307693

Sim(3,1) = 6/(26-5) =   0.2857142857142857

Sim(3,2) = 5/(19-4) = 0.3333333333333333

Sim(3,4) = 5/(16-4) =   0.4166666666666667

Sim(3,8) = 4/(14-3) = 0.36363636363636365

Sim(3,9) = 3/(15-2) =   0.23076923076923078

Sim(3,10) = 1/(12-0) = 0.08333333333333333

Sim(3,14) = 4/(15-3) = 0.3333333333333333

Sim(3,28) = 1/(14-0) = 0.07142857142857142

Sim(3,29) = 1/(13-0) = 0.07692307692307693

Sim(3,33) = 2/(22-1) = 0.09523809523809523

-

Sim(34,9) = 3/(22-2) =   0.15

Sim(34,10) = 1/(19-0) = 0.05263157894736842

Sim(34,14) = 1/(22-0) = 0.045454545454545456

Sim(34,15) = 2/(19-1) = 0.1111111111111111

Sim(34,16) = 2/(19-1) = 0.1111111111111111

Sim(34,19) = 2/(19-1) = 0.1111111111111111

Sim(34,20) = 1/(20-0) = 0.05

Sim(34,21) = 2/(19-1) = 0.1111111111111111

Sim(34,23) = 2/(19-1) = 0.1111111111111111

Sim(34,24) = 4/(22-3) = 0.21052631578947367

Sim(34,27) = 2/(19-1) = 0.1111111111111111

Sim(34,28) = 2/(21-1) = 0.1

Sim(34,29) = 2/(20-1) = 0.10526315789473684

Sim(34,30) = 4/(21-3) = 0.2222222222222222

Sim(34,31) = 3/(21-2) = 0.15789473684210525

Sim(34,32) = 3/(23-2) = 0.14285714285714285

Sim(34,33) = 11/(29-10) = 0.5789473684210527

In that way, similarities of every pair of node from karate network which has 34 nodes, are calculated by applying equation (4.1) and then weight is evaluated according to equation (4.2). When rank with the descending order, the following evaluation can be seen.

Weight of vertex 1=	2.895221119
Weight of vertex 2=	2.654700855
Weight of vertex 4=	2.581296155
Weight of vertex 34=	2.482463735
Weight of vertex 33=	2.475349006
Weight of vertex 3=	2.29037629
Weight of vertex 8=	1.570359053

```
Weight of vertex 14=          1.464646465
Weight of vertex 6=           1.4
Weight of vertex 7=           1.4
Weight of vertex 30=         1.265079365
Weight of vertex 24=         1.214097744
Weight of vertex 9=          1.195054945
Weight of vertex 32=         1.055958747
Weight of vertex 5=          0.909803922
Weight of vertex 11=         0.877674957
--------------------------   ---------------------
            --                        ----
--------------------------   ---------------------
Weight of vertex 22=          0.317647059
Weight of vertex 15=          0.264957265
Weight of vertex 16=          0.264957265
Weight of vertex 19=          0.264957265
Weight of vertex 21=          0.264957265
Weight of vertex 23=          0.264957265
Weight of vertex 10=          0.135964912
Weight of vertex 12=          0.058823529
```

The node with highest weight is 1 and it is selected as initial seed node for finding first local community structure. After the first local cluster is discovered, that cluster includes {1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 18, 20, 22, 10, 17}. The node with highest weight among remaining unassigned nodes into any clusters is node number 34. Therefore, node 34 is selected as the next seed node to produce the next a local cluster.

4.2 System Design

Figure 4.1 shows proposed system design. In this design, seed identification algorithm and community expansion algorithm are used for seed identification phase and community expansion phase, respectively. As an input, social network dataset is used and other networks are also used in this experiment.

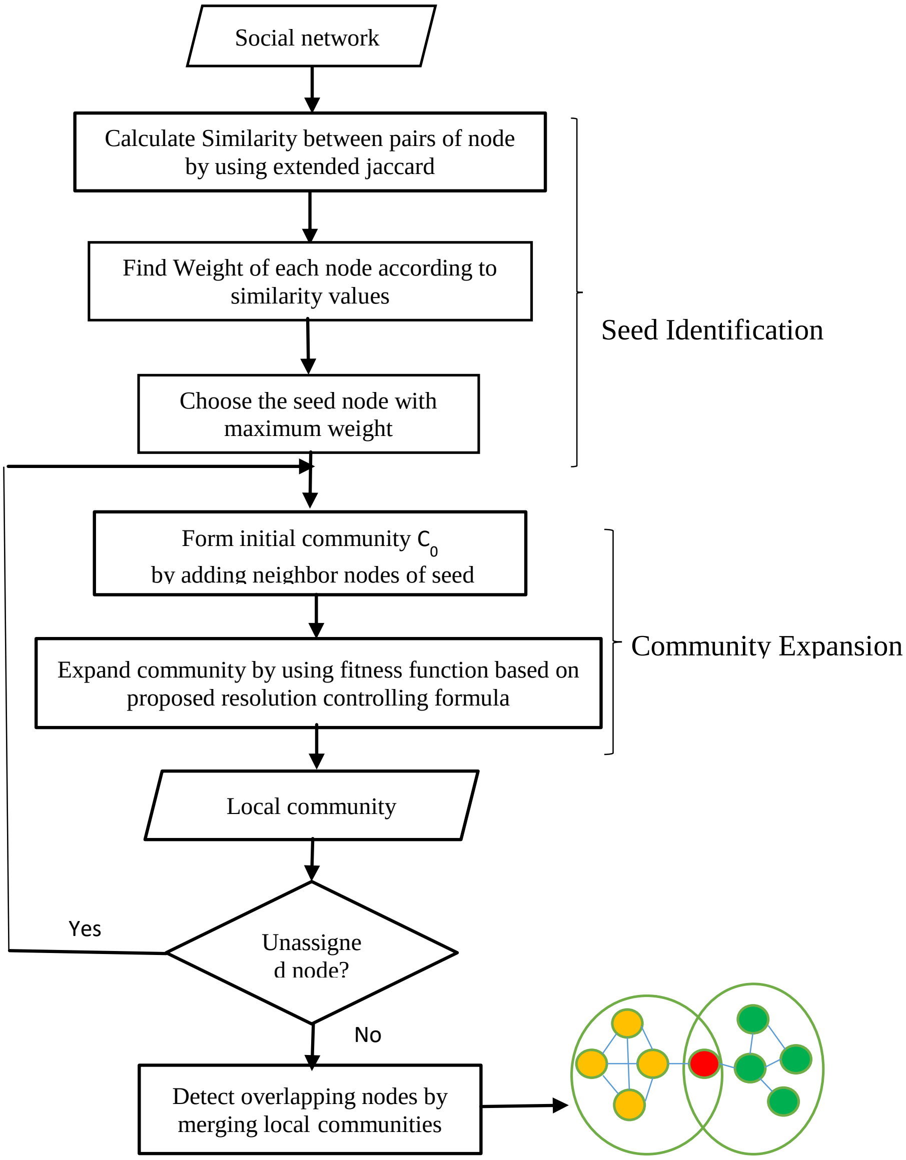

Social network
Calculate Similarity between pairs of node by using extended jaccard
Find Weight of each node according to similarity values
Choose the seed node with maximum weight
Seed Identification
Form initial community C_0 by adding neighbor nodes of seed
Expand community by using fitness function based on proposed resolution controlling formula
Community Expansion
Local community
Unassigned node?
Yes
No
Detect overlapping nodes by merging local communities

Figure 4.1 Proposed system design

4.3 Proposed Algorithms

This section describes algorithms, seed identification and community expansion that are applied in system design of this dissertation. The seed identification algorithm finds core nodes among the nodes as the first phase. Then the community expansion algorithm which extends community to form local community in second phase is explained.

4.3.1 Seed Identification Algorithm

The seed node is a crucial node in forming the local community among all nodes. Initially, the similarity between all pairs of nodes in the entire network is computed using equation 4.1. Then, the weight of each node is determined based on equation 4.2.The example similarity and weight evaluation has already explained in section 4.1.4. The algorithm selects the node with the highest weight as the seed, which is then assigned to the initial community. Algorithm 1 provides the pseudo code for identifying the seed.

Algorithm 1. Seed identification pseudo code

```
Input: G (V,E)
Output: seed
seed=0; weight=0;
for each $v \in V$
    calculate similarity Sim (v , u) of each node by
    extended jaccard similarity;  // $u \in$ neighbor of v
    // compute the weight of each vertex
    weight (v)= $\sum$ Sim (v , u);
end for
seed= max{weight(v)};
return seed;
```

After this process of Algorithm 1, the seed is identified, and the initial community is discovered by considering the neighboring nodes surrounding the seed. Then community expansion process from Algorithm 2 will be continued. As case study, the overlapped objects detection process is illustrated on karate network in section 4.3.3 corresponding to the phase of system flow.

4.3.2 Community Expansion Algorithm

After accepting seed, initial community is formed with neighbor nodes of seed. For extending community, firstly fitness quality of initial community is calculated by using Eq. (6). Subsequently, the fitness of each member within the initial community is evaluated to determine whether they should be removed or retained. If the fitness of the initial community is higher than the fitness of a member node, then that member is removed from the initial community. Conversely, if the member's fitness is equal to or greater than the fitness of the initial community, they are retained in the initial community.

Algorithm 2. Community expansion pseudo code

```
Input:  Initial community $C_0$ , G(V,E)
Output: Local community C
C=∅;
for each v ∈ $C_0$
    calculate fitness value according to f fitness function;
    if F ($C_0$\v) >= F($C_0$ ) then C= $C_0$ – v;
    end if
end for
for each u ∈ $N(C)$ // neighbor nodes of C
    calculate fitness value according to f fitness function;
    if F(C ∪ $u$) > F(C) then C=C ∪ u;
    end if
end for
return C;
```

For example, consider an undirected graph with 12 nodes and 16 edges. For applied undirected graph, the edges are 34. Its average degree is 2 and density is 0.26. After seed identification process, seed = 11, initial community set {7, 8, 9, 10, 11}, internal links within community=7 and outer links =2, α = 2.7. The initial community's fitness, $f_c = 7/9^{2.7} = 0.02$. If remove node7 from community set, fitness of that community is $f_{c-7} = 5/7^{2.7} = 0.03$. In such way, fitness values by extracting each node within community are $f_{c-8} = 5/9^{2.7}$ =0.013, $f_{c-9} = 4/9^{2.7}$=0.01, $f_{c-10} = 4/9^{2.7}$=0.01. Except node 7, quality value of the rest nodes has decreased when each node is removed from community, the node 7 is taken out from initial community. Thus, the current community set is {8, 9, 10, 11}. If this step is finished, then neighbors of community are considered to extend the community. Therefore, fitness of each of neighbor nodes are calculated and node with higher fitness is added to the community. This process continues until the all neighbors of community satisfy the fitness value.

After this process, a local community is obtained. This process is described in Algorithm 2. After obtaining a local community by extending the community, check if there are unassigned nodes to any communities. If it exists, a node with highest weight from remaining nodes is selected as next seed node to form next local community as second iteration. If there are no unassigned node to any community, finally, the overlapped nodes are uncovered when merging the local communities.

4.3.3 Case Study

To show the community discovery results from algorithm, karate network is applied. Firstly after seed identification process, seed node 1 is discovered as center and initial community with neighbors of seed. It is described in Figure 4.2 and Figure 4.3 shows uncovered first local community after expansion process. Then check if there are no unassigned nodes to any community. Because nodes are still left, the second iteration will be done.

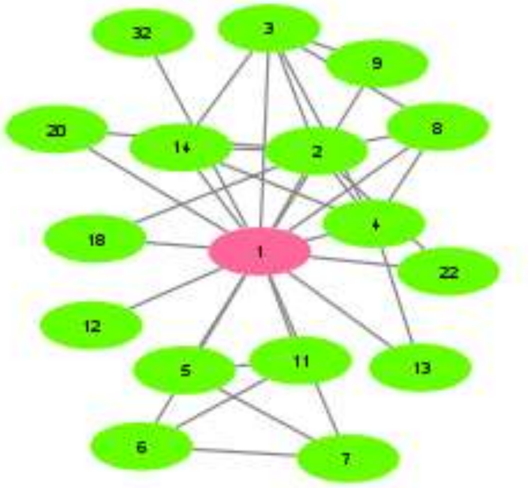


Figure 4.2 Initial community including neighbors surrounding seed for karate

In second iteration, the next seed with the highest weight among unassigned nodes is chose. Therefore, the next seed is 34 and is described with red color. Figure 4.4 illustrates explored initial community from seed identification process for next iteration. The detected local community at the end of community extension phase is shown in Figure 4.5 for seed node 34. Finally, two overlapped community is occurred and two overlapping nodes with gray color are discovered in Figure 4.6.

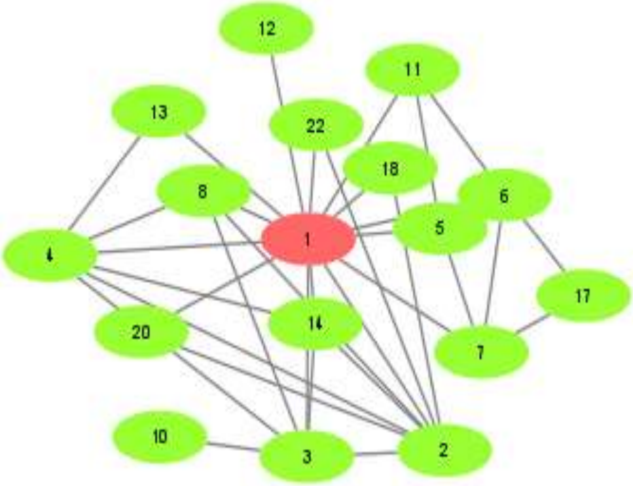


Figure 4.3 The discovered local community after expansion phase for karate

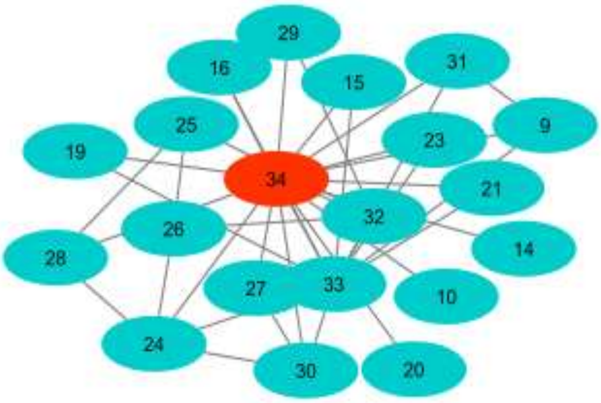


Figure 4.4 Initial community involving seed 34

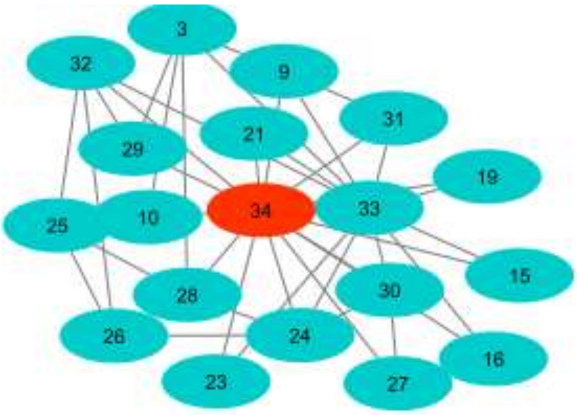


Figure 4.5 The discovered local community after expansion process

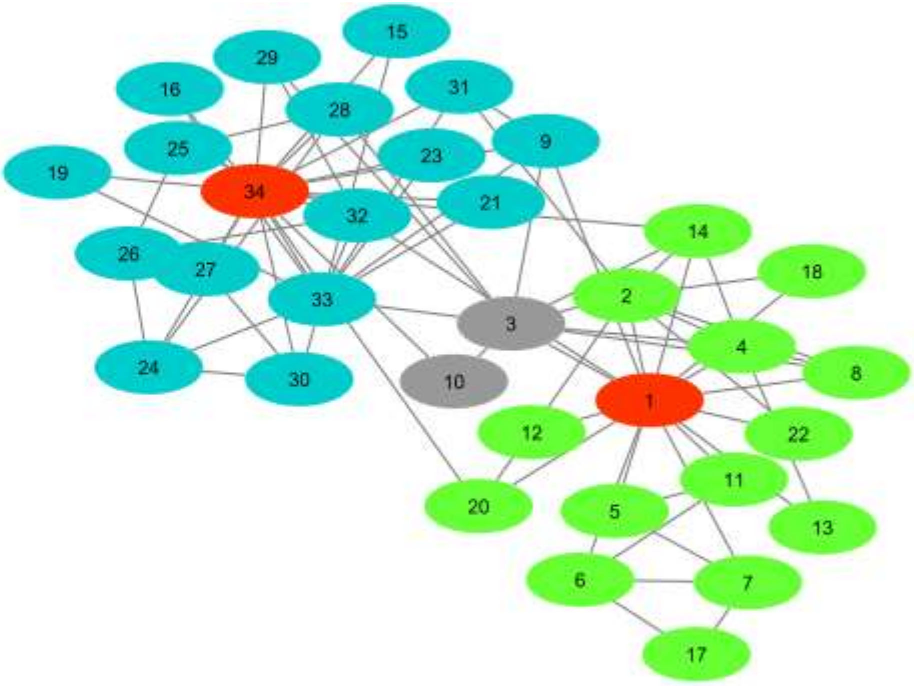


Figure 4.6 The final discovered overlapped communities of karate

4.3 Chapter Summary

This chapter provides the detail explanation of proposed system design and methodologies. The two proposed algorithms, the seed identifying algorithm to detect seeds and community expansion algorithm to find local community are also discussed with case study.

CHAPTER (5)
OVERLAPPED COMMUNITIES DETECTION AND PERFORMANCE RESULTS

This chapter describes the implementation of proposed system and the performance of the proposed algorithm along with the experimental results. Various quality evaluation metrics are used to prove cluster quality of proposed algorithm over other algorithms. For these measurements, real world networks and synthetic networks datasets are applied on the experiments and the performance results are analyzed by comparing other overlapped detection algorithms. The proposed system can partition network into clusters with good quality on not only the real datasets but also synthetic graph datasets. Although the system performs well in detecting overlapping community at community level, less accuracy of overlapped nodes at nodes level on real datasets because ground truth overlapped nodes of real networks are formed by relying on assumption of data collectors. Therefore, the system is tested on synthetic graphs which are generated by program of LFR using all measurements and overlapping fraction. The assessment of overlapping fraction is compared with actual overlapping fraction. Moreover, performance comparisons of extended jaccard and traditional jaccard similarity are also implemented in this chapter.

5.1 Datasets

The experiment results are carried out with real world networks datasets and explain their descriptions in this section. This chapter demonstrates the effective performance results of the system with many evaluation measurements on real datasets and synthetic or artificial graphs. In the experiment, both real and synthetic networks are used to measure system's performance by using ground truth and without ground truth information.

5.1.1 Real Networks

In this experiment, well-known real datasets are applied and they are extensively employed in the field of community detection. The real networks are obtained from different domains and exhibit varying scales and degree distributions.

Karate club network: is a well-known and widely studied example of a social network that has been used in many community detection studies. The data was collected by Wayne Zachary in 1977, and it represents the social interactions between members of a university karate club. The network consists of 34 nodes (representing individual members of the club) and 78 edges (representing friendships or other social ties between members). After a conflict between the administrator of the club and the club's instructor, the network was split into two communities, with one community following the instructor to a new club.

Dolphin social network: The dolphin social network, as compiled by Lusseau et al. in 2003, is another well-known and frequently studied example of a social network in the field of community detection. The network consists of 62 nodes (representing individual dolphins) and 159 edges (representing frequent associations between pairs of dolphins). The dolphins were observed over a period of several years in a community living off Doubtful Sound, New Zealand. The network is undirected, meaning that the associations between dolphins are not directional.

American Football network: represents the games played between Division IA colleges during the regular season of fall 2000. Each node in the network represents a college football team, and the edges represent the games played between pairs of teams. The nodes are also labeled with values that indicate which conferences they belong to. The conferences are groups of teams that compete against each other more frequently than against teams from other conferences.

US politics books network: This network consists of books focusing on US politics that were published during the 2004 presidential election and were available for sale on the online bookstore Amazon.com. The edges between books represent frequent instances of customers purchasing those books together. Nodes have been given values "l", "n", or "c" to indicate whether they are "liberal", "neutral", or "conservative".

Risk map network: this network is a map of the popular strategy board game, Risk and two to six players can play on a board. It is a political map of the Earth, divided into 42 territories, which are grouped into six continents. Therefore, this network comprises 42 vertices and 83 edges in accordance with the six continents.

Ego_facebook: this network contains Facebook user–user friendships. A node represents a user. An edge indicates friend relationships of user to user. Facebook data was collected from survey participants using this Facebook app.

Net science: It is a network of coauthor ships between scientists 1588, scientists in this case who are themselves publishing on the topic of networks. It is based on data, including publications up until early 2006. In this network, multiple nodes represent authors and edges represent joint publications on which authors have collaborated together.

Amazon product co-purchasing network: represents products sold by Amazon.com, and the edges between products represent frequent co-purchasing of products by the same buyers. Each product category defines a ground-truth community, meaning that products that are frequently purchased together are likely to belong to the same category or have similar properties.

DBLP collaboration network: represents collaborations between authors who have published papers in computer science journals and conferences. Nodes in the network represent authors, and an edge between two nodes indicates that the corresponding authors have co-authored at least one paper together. The ground-truth communities in this network are defined based on the journals or conferences where papers were published.

Table 5.1 Statistics of real datasets

	Node	Edge	density	Avg. degree	Ground truth
Karate[101]	34	72	0.14	4	Y
Dolphin[102]	62	159	0.08	5	Y
Political book[105]	105	441	0.08	8	Y
Football [22]	115	613	0.09	10	Y
Riskmap[105]	42	83	0.096	3	Y

Ego-facebook [104]	2888	2981	7.15 E-4	2	N
Netscience[105]	22963	48436	0.00257	3	N
DBLP [106]	317080	1049866	0.001196	6	Y
Amazon [106]	334863	925872	0.004	5	Y

5.1.2 Synthetic Networks

The experiment also employs the LFR benchmark [18] to create synthetic datasets. The network created by this program has the ability to accurately regulate the distribution of node degrees and community sizes. LFR is proposed by Lancichinetti and Fortunato and is a type of computer-generated networks with predefined tunable parameters. The networks possess real world characteristics and are widely applied in various overlapping network community detection algorithms. The basic parameters are illustrated in Table 5.2. This experiment sets the network parameters as follows: N = 1000 to 50000, k= 15, μ= 0.3, on= 10%, om= 2, maxk=50, minc=10, maxc=50;

Table 5.2 Meaning of parameters

Parameter	Meaning
N	Number of nodes
k	Average degree of nodes
maxk	The nodes’ maximum degree
maxc	Nodes’ maximum cluster size
minc	Nodes’ minimum cluster size
mu (μ)	Mixed parameter
om	The number of communities to which overlapping nodes belong
on	Number of nodes which belongs to multiple communities

5.2 Performance Analysis

This section showcases the outcomes achieved by implementing the techniques outlined in chapter 4 on a collection of 9 actual networks that are commonly utilized as benchmarks in network community structure studies. All the experiments were carried out using Java on a PC laptop equipped with an Intel Core i5 processor running at 2.7 GHz, a 64-bit CPU, and 8 GB of RAM. The laptop was running Windows 10. The large real datasets have taken from standford network dataset collection site and some small data are available at personal homepage server for the University of Michigan.

In this dissertation, the performance of proposed system is analyzed by using evaluation metric with both ground truth and no ground truth information. The evaluation metrics have been described in section 3.9 of chapter 3. That section has presented widely used measurements for overlapping communities on both ground truth information and measurements without ground truth. ENMI (Extended Normalized Mutual Information), Omega index, F1 measure are applied with ground truth and Qov is used with no ground truth. What is more, overlapping rates are compared with other state of the art overlapping detecting algorithms. The following algorithms are used as based line algorithms in this part. DEMON, LFM, NILPA, OSLOM and GREESE are Democratic Estimate of Modular Organization of a Network, Local Fitness Method, Node Important based Label Propagation, Order Statistics Local Optimization Method and Greedy Coupled-seeds Expansion, respectively. They are analyzed on both real datasets and benchmark (artificial) networks consisting of N= (1000, 2000, 3000, ----, 5000). On benchmark networks, running time of proposed algorithms is also illustrated and the efficient running time of proposed system has occurred over the other algorithms.

5.2.1. Evaluation Results for Real Networks

Based on the ENMI (Normalized Mutual Information Entropy) results in Figure 5.1, the proposed system demonstrates superior performance compared to other algorithms in the karate, Amazon, and DBLP networks. In the political book network, the proposed system shows a slightly better performance than the others. NILPA performs well on small and sparsely connected networks like risk-map but it cannot be partitioned the football network into communities by resulting in only one community being identified with a measurement of zero. It does not perform effectively on medium and large networks according to the results. The OSLOM algorithm achieves a higher ENMI value than all other algorithms on the football network, while GREESE outperforms the proposed algorithm significantly on the dolphin network. The proposed algorithm decreases the accuracy in dolphin, football, political and riskmap datasets because the ground truth communities of these datasets are formed based on their features and natures. The football and political book datasets have node features, and the ground truth of the football network, for instance, it is based on conferences to which teams belong.

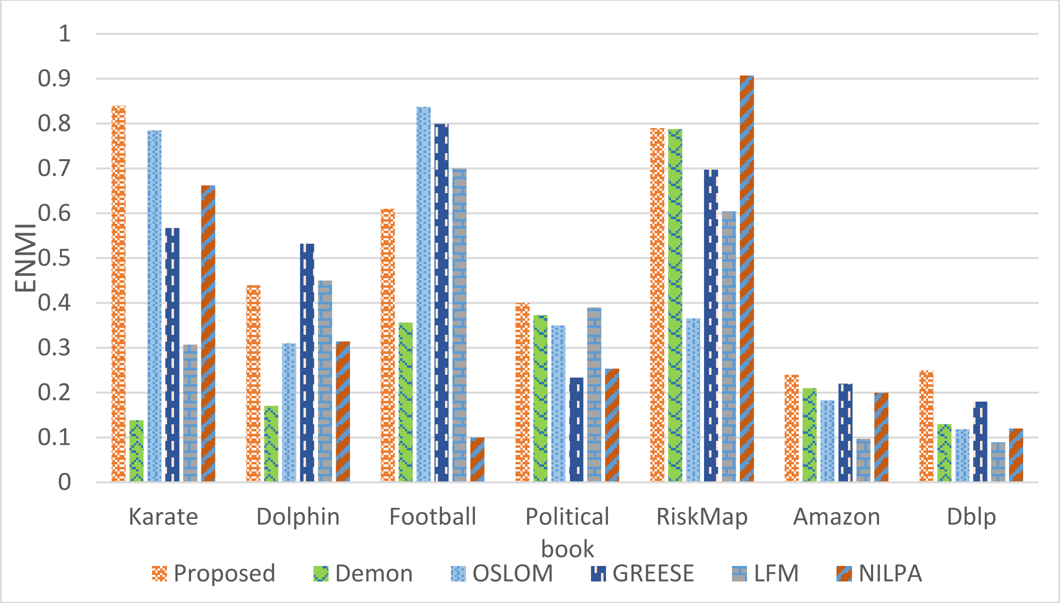


Figure 5.1 Comparison results of ENMI on different datasets

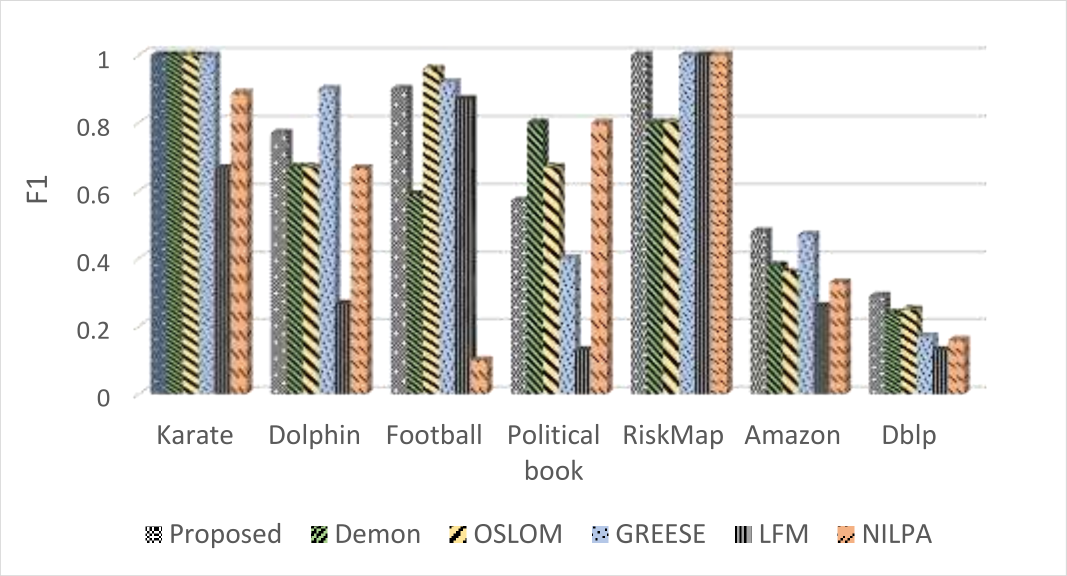


Figure 5.2 F1 results of comparison algorithms

Furthermore, the large networks such as Amazon and DBLP show slightly better outcomes according to the F1 results of the proposed system in Figure 5.2. All algorithms, except for LFM, produce similar results in the karate and risk map networks. There are no significant differences observed in this experiment. In this measurement, similar to the ENMI measurement, GREESE performs better on the dolphin network, while OSLOM demonstrates superior performance on the football network. However, OSLOM generates the isolation nodes and it cannot be added all nodes from the graph to the corresponding communities. As for NILPA, the F1 evaluation for the football network is zero, indicating the absence of communities.

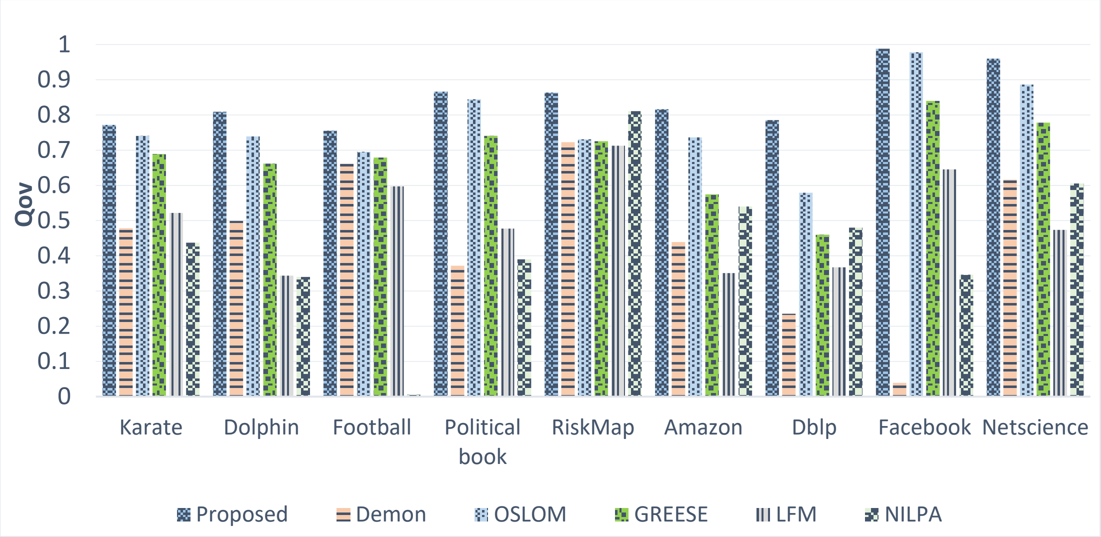


Figure 5.3 Qov comparison of overlapping algorithms

The evaluation of the quality of overlapped communities is measured using a metric called Qov, without relying on any predefined ground truth. This evaluation is conducted on various datasets, disregarding the presence of ground truth. The quality of proposed system is observed to surpass that of other algorithms across all datasets. When comparing the baseline methods, including the proposed algorithm, it is found that LFM decreases accuracy in all measurements which is shown in Figure 5.3.

The performance comparisons of the proposed extended and traditional jaccard similarity are illustrated in from Figure 5.4 to 5.6. According to these experiments, it is seen that there is beneficial results in the extended jaccard similarity because of that results, it is important to identify core nodes for each local community in local community discovery methods. To be continued, the overlapping fraction is described in this dissertation.

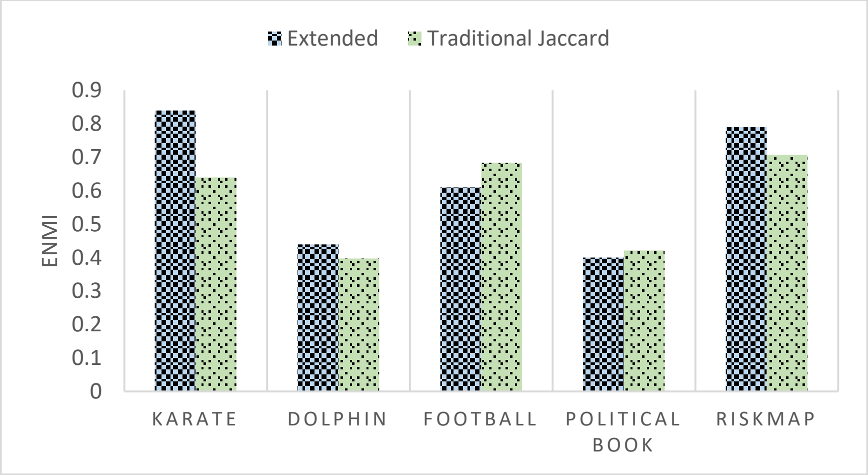


Figure 5.4 ENMI of extended and traditional jaccard on real networks

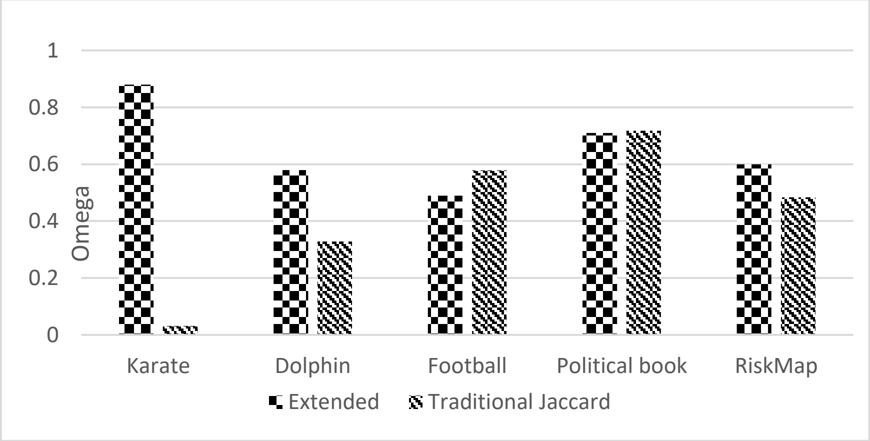


Figure 5.5 Omega index of extended and traditional jaccard on real networks

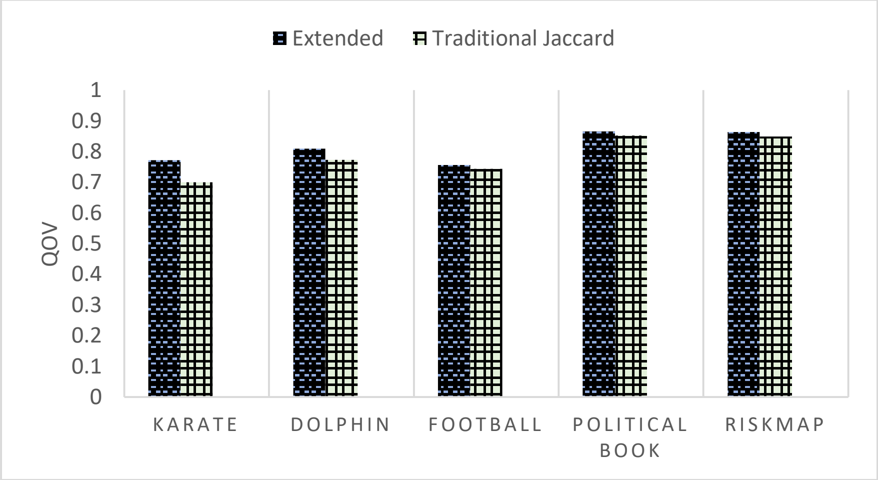


Figure 5.6 Qov result of real networks

In Figure 5.7, the proposed system exhibits an acceptable level of overlapping in the communities it discovers. This rate is calculated as a percentage, representing the ratio of the number of overlapped nodes within the discovered communities to the total number of nodes in the graph. When there is an excessive number of overlapped nodes, the system generates hierarchical communities, resulting in numerous highly hierarchical structures. The Demon method tends to have an excessively high fraction of overlapping, leading to isolated nodes. Similarly, NILPA demonstrates a high overlapping rate in networks other than the football network. However, this method fails to explore overlapped nodes specifically in the football network. OSLOM and LFM are unable to identify overlapped nodes in the football and karate networks, respectively, despite having a low overall overlapping ratio in some networks. To address issues related to isolated nodes and excessive overlapping, the GREESE method refines the community structure.

The proposed system does not occur outlier and can detect, in addition to, all nodes from the network to the corresponding communities. In order prove whether acceptable overlapping rate is or not, this overlapped fraction is evaluated on LFR benchmark networks. The parameters for generating that network are set to N=1000 to 5000, k=15, mu=0.3, om= 2, on=10% on N, maxc=50, minc=10, maxk=50.

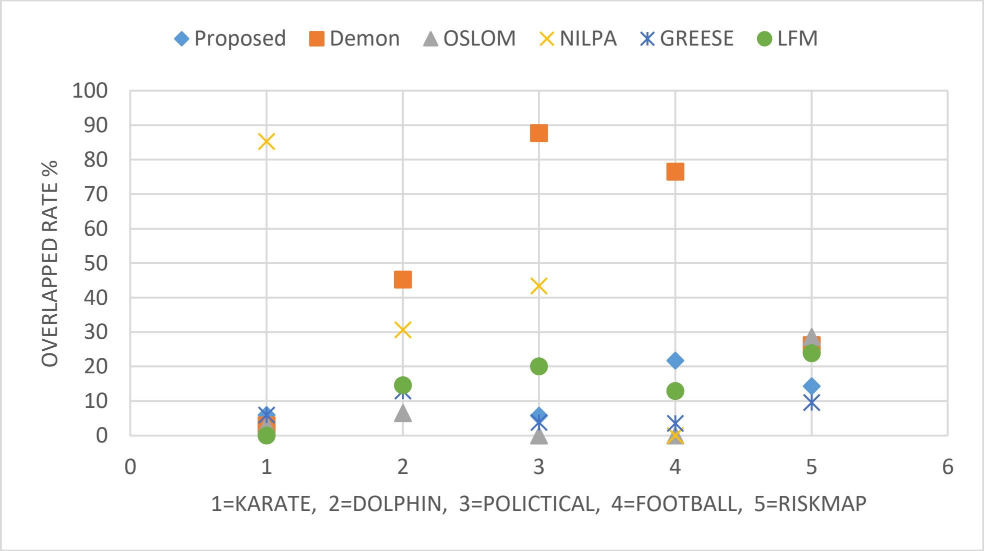


Figure 5.7 Overlapping rate of different algorithms

5.2.2 Evaluation Results on Benchmark Graphs

This part analyzed the results of proposed algorithm with baseline methods on benchmark (artificial) graphs including number of nodes 1000 to 5000. The parameter setting to generate graph has been described in the above section 5.1.2 and notation description of that parameter has been listed in Table 5.2.

According to the ENMI performance of Figure 5.8, the proposed algorithm has better accuracy than LFM, Demon, and GREESE. With OSLOM and NILPA, proposed methods have almost same result on all LFR networks and each of them is only 0.01 apart. OSLOM and NILPA can perform well on benchmark graphs but they cannot perform well in real networks. GREESE can reveal good overlapped community structures on only real networks. LFM does not have good ENMI result in both real and benchmarks.

F1 results also perform well on the benchmarks like OSLOM and NILPA. Although the proposed method cannot outperform on all real datasets, it has good quality overlapped structure on some real datasets and LFR benchmarks when compare other overlapped detection algorithms. F1 evaluation result is demonstrated in Figure 5.9. With both ENMI and F1 results, OSLOM and NILPA seem to perform well because these methods based on label propagation strategy. Therefore, they have slightly increased in accuracy but running time of that methods takes a long time. LFM algorithm cannot detect overlapping nodes accurately on 3000 nodes. Therefore, its' ENMI and F1 results are zero.

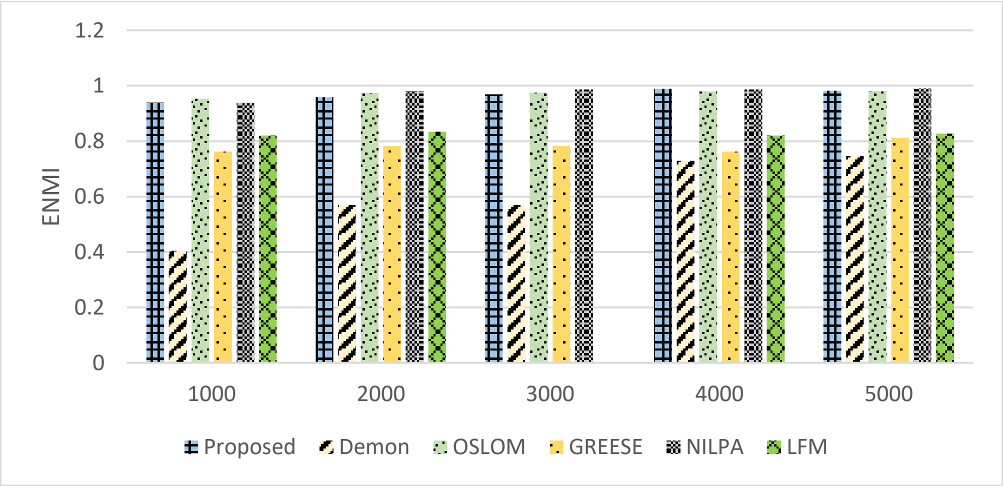


Figure 5.8 ENMI of different algorithms on overlapping LFR benchmark

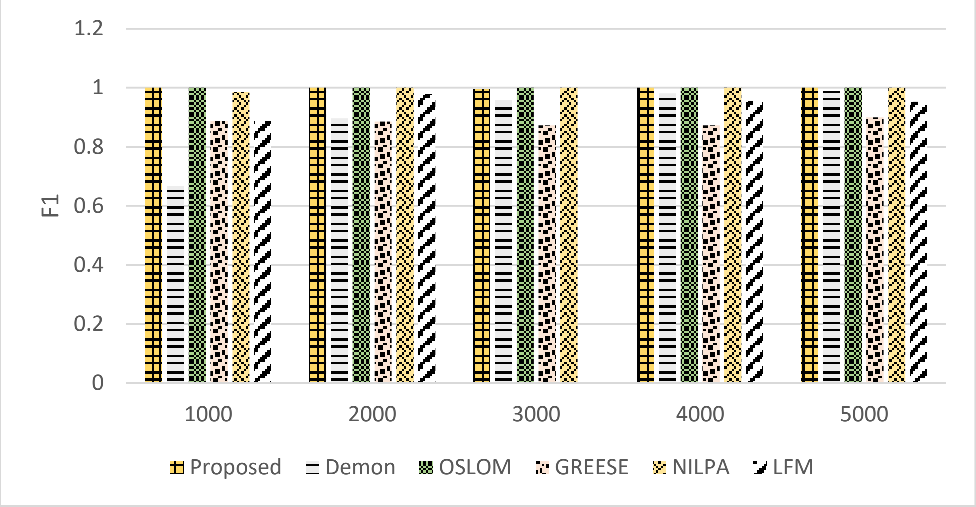


Figure 5.9 F1 result comparison on LFR benchmark

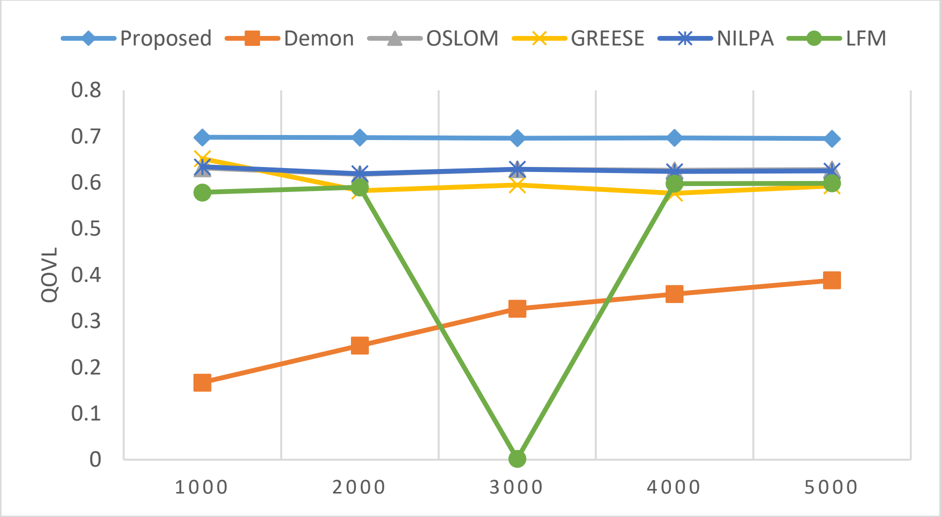


Figure 5.10 Qov of comparison algorithm for LFR

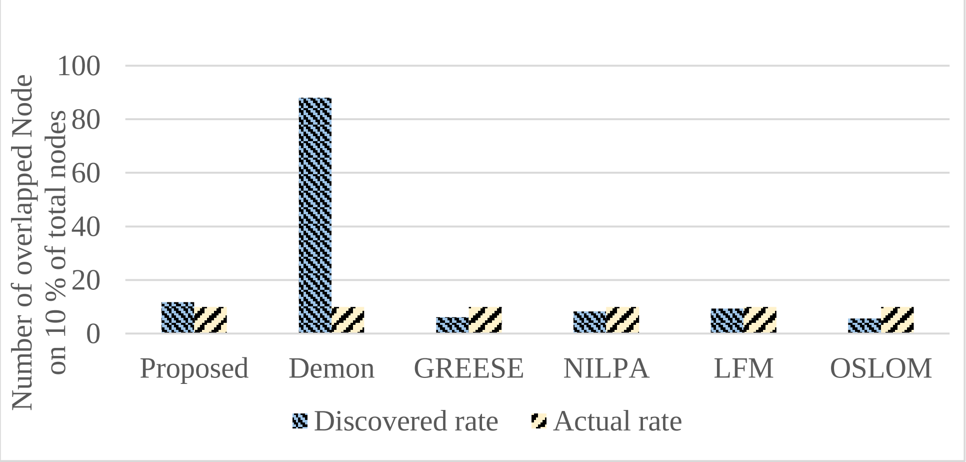


Figure 5.11 Overlapping rate of comparison algorithms on N= 1000

Figure 5.10 shows the overlapping modularity (Qov) result. The proposed method outperforms over the comparison algorithms in measurement of Qov which measure quality of overlapped structure with no ground truth information.

To verify appropriate overlapping rate of discovered overlapping structure, percentage of overlapped nodes is set to 10% of total number of nodes of graph as parameter when generate the LFR benchmark graph. Therefore, number of overlapped node is set to ten, twenty, thirty, forty and fifty nodes on 1000, 2000, 3000, 4000 and 500 respectively. The overlapping fraction of comparison algorithms on each LFR graph is illustrated in Figure 5.11 to Figure 5.15. With respect to Figure 5.11, the detected number of overlapping nodes is 10 and overlapped fraction is 10% on 1000 nodes. Therefore, it is found that overlapping accuracy is at good result. LFM and NILPA have also competitive result but LFM occur isolation nodes and sometimes, it cannot detect overlapping nodes on 3000 nodes. Therefore, LFM may have good result if it is compared on overlapped rating but it cannot identify overlapped nodes accurately when compare performance quality at overlapped node level by observing ENMI and F1.

According to evaluation results of overlapping rate, the rate of discovered overlapping nodes from proposed algorithm can discover exactly on node 4000 and 5000. In node 1000 and 2000 data size, it can detect approximate overlapping rate with the actual rate. The ground truth data of real world networks do not have overlapping rates and some networks have also no overlapped nodes. Therefore, the overlapping node parameter is set to 10% for the overlapping community structure generated by LFR and the proposed algorithm is tested.

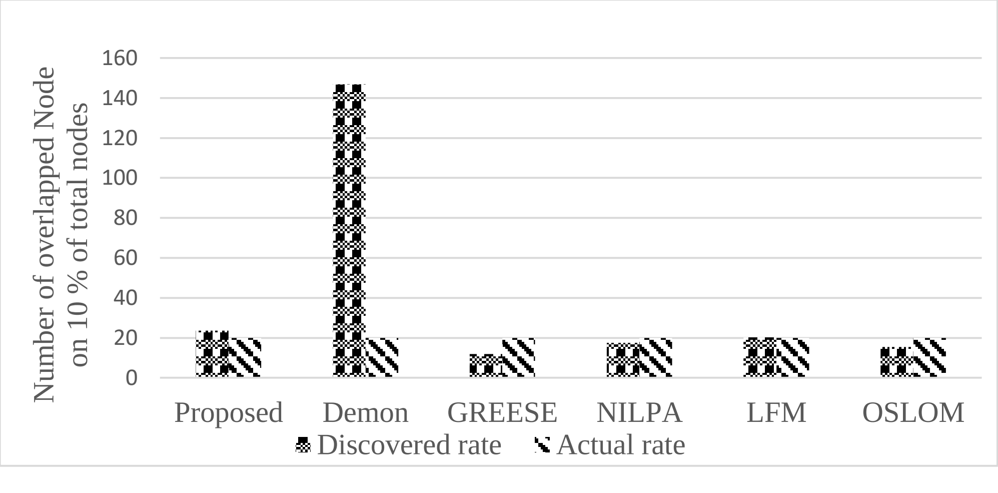


Figure 5.12 Overlapping rate of different algorithms on N=2000

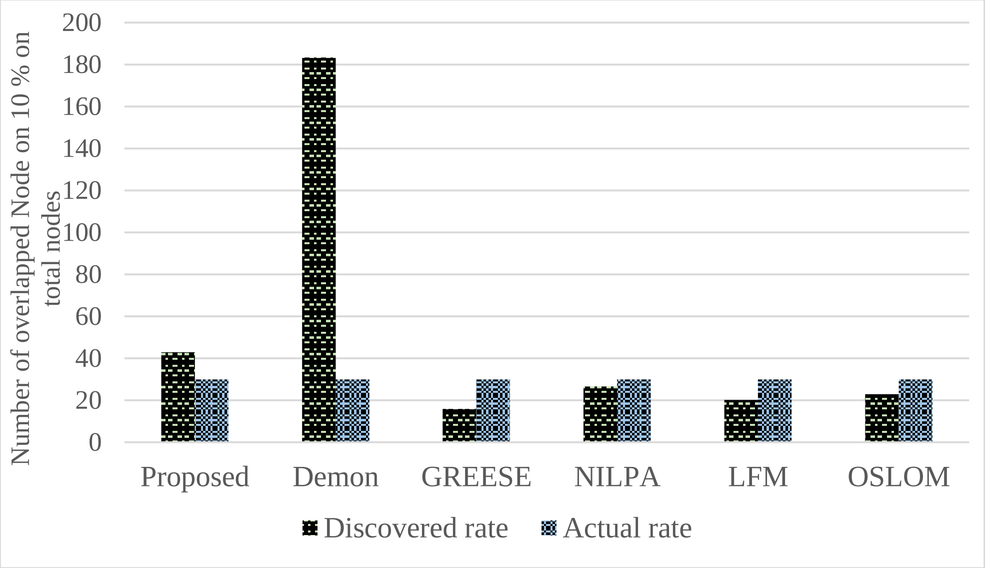


Figure 5.13 Overlapping rate of N= 3000

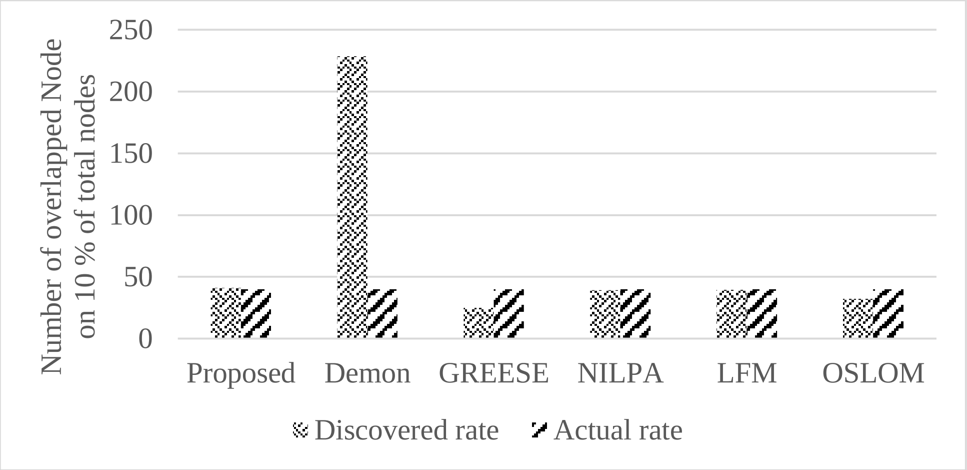


Figure 5.14 Overlapping rate of algorithms on N=4000

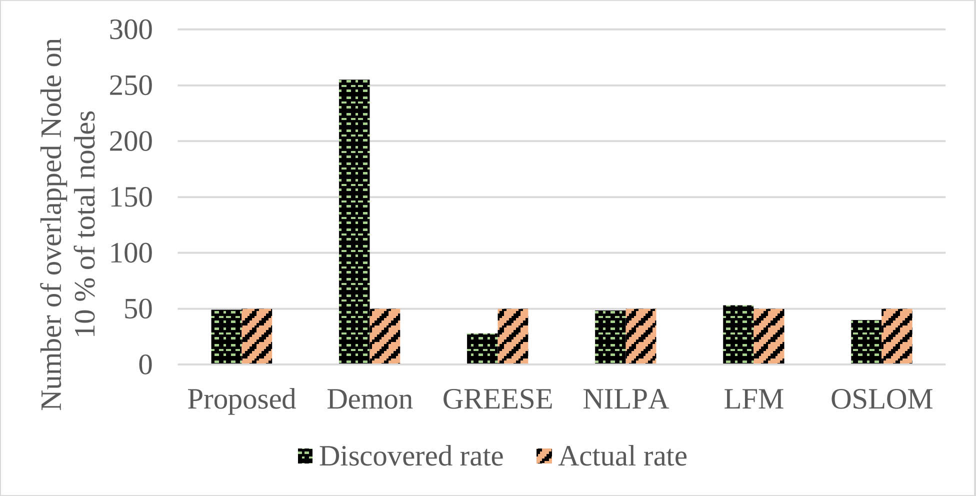
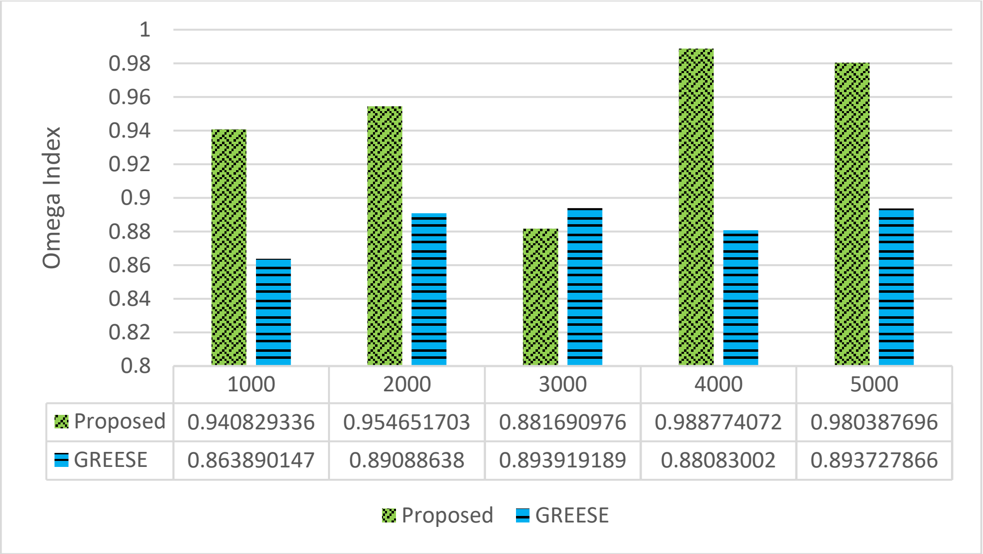


Figure 5.15 Overlapping rate of different algorithms on N= 5000

Omega results are analyzed by comparing proposed method and GREESE in the Figure 5.16. Except N=3000 network, the result of proposed algorithm can compare GREESE in other LFR networks. The Figure 5.17 to 5.19 are also shown as performance evaluation for extended and traditional jaccard similarity on benchmark graphs. In contrast, the extended jaccard outperforms than traditional in all measurements of all benchmark graphs.

As shown in Table 5.3, the run time comparison of overlapping detection methods is described in millisecond. The proposed algorithm is the faster than the others. NILPA and OSLOM have accurate result on LFR benchmark graphs corresponding to their performance evaluation metrics but its execution time take more time. Not only in accuracy but also in execution time, the method is competitive in benchmark graphs.



	1000	2000	3000	4000	5000
▨ Proposed	0.940829336	0.954651703	0.881690976	0.988774072	0.980387696
☰ GREESE	0.863890147	0.89088638	0.893919189	0.88083002	0.893727866

Figure 5.16 Omega measurement on LFR networks

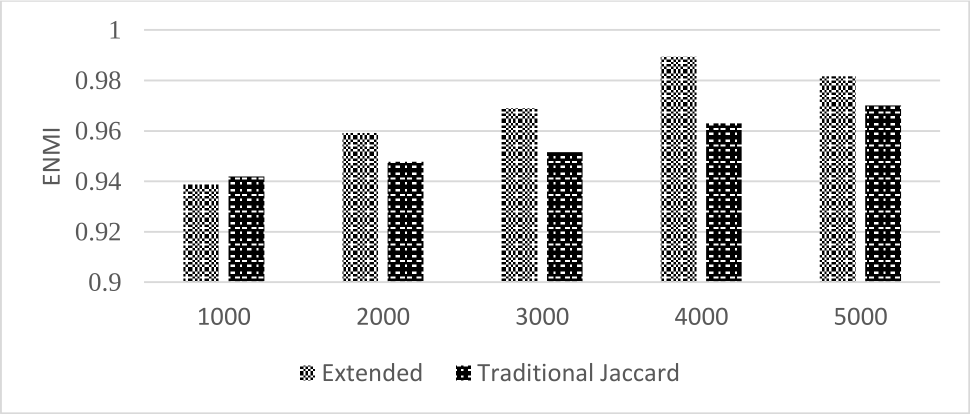


Figure 5.17 ENMI of extended and traditional jaccard on LFR networks

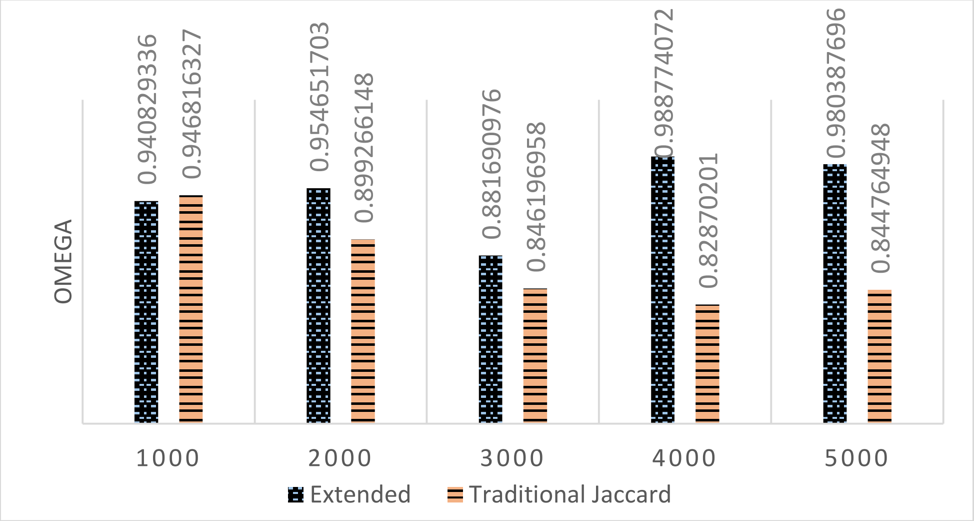


Figure 5.18 Omega index on LFR networks

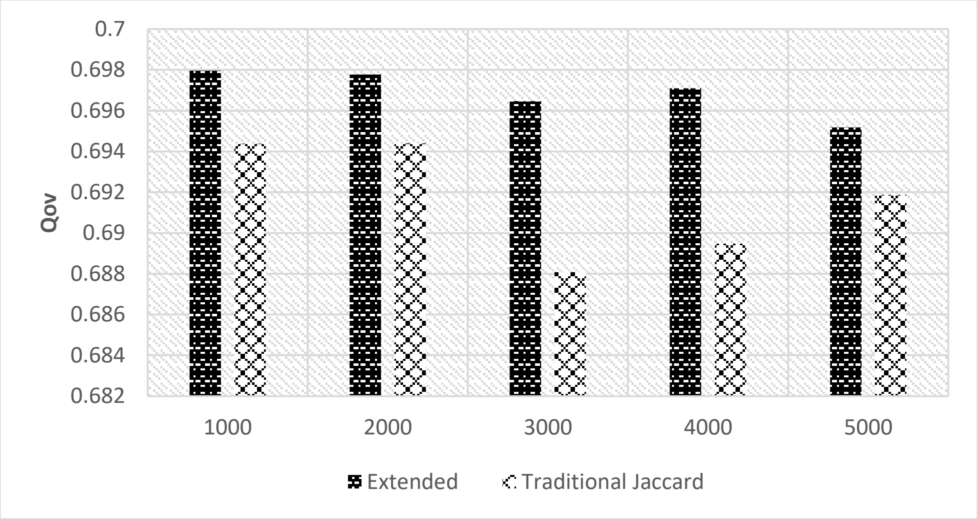


Figure 5.19 O_{ov} measurement of LFR networks

Table 5.3 Running time comparison in milliseconds

Nodes	Proposed	Demon	GREESE	NILPA	LFM	OSLOM
1000	0.612	2100.012	716.211	62.820	32100.002	34560.589
2000	0.650	3840.051	2140.256	190.122	36015.210	40027.481
3000	0.831	6900.452	4650.012	356.119	55042.008	72146.270
4000	1.108	8740.015	7740.002	747.023	89071.056	100245.780
5000	1.314	11270.120	11930.11	1747.080	102045.054	250786.110

5.3 Chapter Summary

In this chapter, the experimentation and evaluation of the proposed algorithm by using various quality assessment methods for overlapped communities. The experiments compared the state of the art overlapping algorithms with various quality measurements by applying real world datasets and benchmark datasets (artificial networks). Overall, the proposed algorithm is found to be better on all measurements for most datasets especially large scale network datasets and it takes less running time than the other algorithms. Although the proposed extended similarity method is not occurred significantly difference on some measurements for real networks, it outperforms on benchmark graphs.

CHAPTER (6)
CONCLUSION AND FUTURE WORK

Community analysis of a network has been a consistent area of focus among researchers for the past decade and a half. Much of the research has aimed to develop algorithms for identifying and detecting communities within networks. Real world systems are made of elements with complex interconnections in between them. The biochemical networks, human networks, collaboration networks are examples of biological, social, and scientists' paper collaboration systems, respectively. The distinction in characteristics between communities in real networks, such as non-overlapping and overlapping communities, has posed a challenge when using existing algorithms for community detection. Therefore, although exposing the overlapped community structure has increasingly been interested, the most researchers ignore these features and focus on the disjoint community structures. This research focuses on uncovering overlapped community structures and local community discovery method is applied to discover overlapped objects instead of global community discovery methods.

6.1 Conclusion of Research

This work proposes an algorithm for detecting overlapping communities through a local community expansion strategy. Most existing overlapping community detection algorithms employ a local expansion strategy lead to instability in the community structures due to fluctuations in the fitness evaluation function they utilize. The detected community structure heavily relies on a parameter that controls the resolution of the communities. Hence, this study presents a parameter evaluation formula that prioritizes resolving the instability of community structures to mitigate the impact of parameter choices. To ensure efficient computation time, the local community expansion algorithm is designed by optimizing the fitness evaluation function, f, based on the parameter evaluation formula. Additionally, the extended Jaccard similarity is employed to determine the seed node, as the core node or seed plays a crucial role in identifying the center of the local community.

In the experiment of this dissertation, both real world networks and LFR benchmark datasets are applied. Some real world datasets do not include ground truth information. For datasets consisting of ground truth, therefore, the performance of the proposed algorithm is evaluated by ENMI, Omega index and F1. For datasets with no ground truth, evaluated by Qov. The results show that, Qov measurement of the proposed algorithm has the significant improvement on both type of datasets and overlapped fraction is not high. Moreover, other performance results (ENMI, F1) are occurred with better accuracy on benchmark datasets, while on real datasets it only achieves more accurate results on most of the datasets. In addition, it saves more running time than the others on both real and benchmarks.

6.2 Research Discussion

By analyzing based on 9 real network datasets including small, medium and large size, it is found that the proposed system has better accuracy than the other base line algorithms (Demon, OSLOM, LFM, GREESE, NILPA) on the most of the datasets. OSLOM and GRESSE methods have also competitive results on ENIM and F1 measurements that measure by using ground truth information. However, in large complex networks, the proposed method has more effective result than the other algorithms including OSLOM and GREESE according to ENMI, F1 and Qov measurements. What is more, the overlapped modularity (Qov) of proposed method outperforms over the all. That is why, satisfied results are occurred by system if see the various quality evaluation results. According to the overlapping fraction evaluation, proposed system and GREESE have appropriate overlapping rate. Except those two methods, the others cannot detect overlapped nodes for some datasets and they generate isolation nodes which cannot assigned to any communities. Therefore, omega index results have not been described in the comparison with real datasets.

When implementing algorithms with 5 LFR benchmark graphs including node 1000 to 5000, the proposed method can reveal overlapping community structure at better accuracy. Just like when experimented in real networks, the OSLOM has competitive results on benchmark graphs. However, the execution time of OSLOM is long. NILPA has been occurred that it has accurate performance results on these graphs even it has poor accuracy in real graphs. The proposed system performs well with all quality measurements on LFR networks. In addition, performance of extended and traditional jaccard similarity is measured on both real and LFR networks. With results from experiments, it is found that extended jaccard has good results except football and political book networks because ground truth communities of these datasets were formed by relying on their features like political book types. That is why, the members within a community do not directly concerned with the core node. For this reason, the proposed system is a slight decrease in performance for these two real datasets. Although it has decreased a bit in result of these two real datasets, it is found that the all benchmark datasets have greater results on all measurements. Moreover, the proposed system also save more time than other overlapped detection methods in running time.

6.3 Advantages and Limitations

The OSLOM algorithm operates with more execution time although it achieves good accuracy score on both real and benchmarks. In experiments with various evaluation criteria, the proposed method shows good accuracy in real datasets and better accuracy in benchmark datasets. With respect to running time, the proposed algorithm has competitive run time than the other algorithms and it can detected overlapped communities in good efficiency.

The system has been implemented only on homogeneous networks and focus on the links between nodes. Therefore, the features of network such as attributes and weight on links are not considered. As ground truth communities can be different for each dataset depending upon the features of classification. This situation declines the accuracy in some real networks, which form the ground truth communities by relying their attributes. In addition, ground truth of some real networks has no overlapped nodes. In that case, the accuracy for some networks is less than some algorithms when the performance of the overlapped structure is measured by evaluation metrics. Therefore, this dissertation also describes performance of algorithm on the artificial graphs (LFR benchmark) and Qov metric for overlapping communities for increasing the accuracy.

6.4 Future Works

In the real world, the structure of networks undergoes dynamic changes over time. As a result, communities within these networks can also evolve, experiencing growth, contraction, merging, splitting, birth, continuation, and even dissolution. To address these changes in community structure, future studies will focus on developing methods to handle dynamic networks. Furthermore, in recent years, there has been significant development in utilizing deep learning techniques for community detection, particularly for effectively handling high-dimensional network data. (i.e. high dimensional features of heterogeneous networks). The studies like that will be tried to know whether there is significant improvement.

Author’s Publications

1. Eaint Mon Win, May Aye Khine, “A Study on Community Overlapping Detection Algorithms in Social Networks”, 12th International Conference on Future Computer and Communication (ICFCC), pp: 136-140, 2020

2. Eaint Mon Win, May Aye Khine, “Overlapped Community Detection using Extended Node Similarity by Local Expansion”, 17th International Conference on Computer Applications (ICCA), pp: 2021

3. Eaint Mon Win, Si Si Mar Win, “Exploring Overlapped Communities based on Optimized Community’s Resolution Controlling Evaluation in Social Networks”, International Journal of Intelligent Engineering and Systems (IJIES), vol:16, issue 1, 2023

BIBLIOGRAPHY

[1]. S.H. Strogatz, “Exploring complex networks”, Nature 410, 268–276 (2001)

[2]. M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.

[3]. R. Xu, D.C. Wunsch; “Clustering algorithms in biomedical research: a review”, IEEE reviews in biomedical engineering, volume 3, 120-154, 2010, IEEE

[4]. J.P. Bigus; “Data mining with neural networks: solving business problems from application development to decision support”, 1996, McGraw-Hill, Inc.

[5]. G. Mecca, S. Raunich, A. Pappalardo; “A new algorithm for clustering search results Data & Knowledge Engineering”, volume 62, 504-522, 2007, Elsevier

[6]. A. Lancichinetti, S. Fortunato; “Community detection algorithms: A comparative analysis. PHYSICAL REVIEW E 80, 056117 2009

[7]. J. Leskovec, K. J. Lang, M. W. Mahoney; ”Empirical comparison of algorithms for network community detection”, In Proceedings of the 19th Conference on World Wide Web (WWW’10). 631–640, 2010

[8]. A. Clauset, MEJ. Newman, C. Moore; “Finding community structure in very large networks”, Phys Rev E 2004, 70:066111

[9]. P. Pons, M. Latapy; “Computing communities in large networks using random walks”, In: Proceedings of the 20th International Conference on Computer and Information Sciences. Berlin/Heidelberg: Springer-Verlag; 284–293, 2005

[10]. M. Rosvall, D. Axelsson, and C. T. Bergstrom; “The map equation”, The European Physical Journal Special Topics 178, pp: 13–23, (2009).

[11]. S. Kelley, M. Goldberg, M. Magdon-Ismail, K. Mertsalov, A. Wallace; “Defining and discovering communities in social networks”, In Handbook of Optimization in Complex Networks, Springer, 2011, 139–168.

[12]. F. Reid, A. F. Mcdaid, AND N. J. Hurley; “Partitioning breaks communities”, In Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM’11), 2011, 102–109.

[13]. A. Lancichinetti; S. Fortunato; J. Kertész; “Detecting the overlapping and hierarchical community structure in complex networks”, New journal of physics, 2009, IOP Publishing

[14]. H. W Shen, X. Q. Cheng, K.Cai, and M. B. Hu; “Detect Overlapping and Hierarchical Community Structure in Networks”, Physica A: Statistical Mechanics and its Applications, Vol. 388, 1706-1712. (2009)

[15]. M. Waing; S. Yang, L. Wu; " Improved community mining method based on LFM and EAGLE”, Computer Science and Information Systems, vol 13, 515-530, 2016

[16]. H. You, X, Zhang, H. Fu, Z. Zhang, M. Li, X. Fan; " Algorithm of detecting overlapping communities in complex networks", 2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)"

[17]. RV. Belfin, P. Bródka; “Overlapping community detection using superior seed set selection in social networks”, Computers & Electrical Engineering, vol 70,1074-1083, 2018, Elsevier

[18]. A. Lancichinetti, S. Fortunato; “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities”, Physical Review E, vol 80, 2009, APS

[19]. J. Ma and J. Fan, “Local Optimization for Clique-based Overlapping Community Detection in Complex Networks”, 2019, IEEE

[20]. P. Bedi, C. Sharma; “Community detection in social networks”; Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol:6, pp: 115-135, 2016 Wiley Online Library

[21]. M. E. J. Newman, “Detecting community structure in networks”, The European Physical Journal B - Condensed Matter and Complex Systems, 38(2):321–330, Mar. 2004

[22]. M. Girvan, M.E.J. Newman; “Community structure in social and biological networks”, Proc.Natl. Acad. Sci. USA **99**, 7821–7826 (2002)

[23]. M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks”, Physical Review, E 69(026113), 2004.

[24]. B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs”, Bell Sys. Tech. J., 49(2):291308, 1

[25]. P. J. Mucha, T. Richardson and M. A. Porter, “Spectral tri partitioning of networks”, Physical Review Letters E, 2009.

[26]. M. Mutingi, C. Mbohwa; "Grouping genetic algorithms Advances and Applications", Switzerland: Springer International Publishing, vol: 243, 2017, Springer

[27]. K. Öztürk; "Community detection in social networks", 2014, East Technical University

[28]. R. Merris; "Laplacian matrices of graphs: a survey Linear algebra and its applications", vol 197, pp: 143-176, 1994, Elsevier

[29]. M. EJ. Newman; "Finding community structure in networks using the eigenvectors of matrices", Physical review E , vol 74, 2006, APS

[30]. E. Abbe; "Community detection and stochastic block models: recent developments", The Journal of Machine Learning Research, vol 18, pp: 6446-6531, 2017, JMLR. org

[31]. S. Fortunato, D. Hric, "Community detection in networks: A user guide", Physics reports vol: 659, pp: Jan-44, 2016, Elsevier

[32]. SW. Son, H. Jeong, J.D. Noh; "Random field Ising model and community structure in complex networks", The European Physical Journal B-Condensed Matter and Complex Systems, vol 50, pp: 431-437, 2006, Springer

[33]. S. Fortunato; "Community detection in graph", pp: 75-174, 2010, Elsevier

[34]. G. Palla, I.J. Farkas, I. Derényi, T. Vicsek; "Uncovering the overlapping community structure of complex networks in nature and society", Nature 435:814–818,2005

[35]. B. Adamcsek, G. Palla, I.J. Farkas, I. Derényi, T. Vicsek; "Cfinder: locating cliques and overlapping modules in biological networks", Bioinformatics 22:1021–1023,2006

[36]. Y.Xing, F.Meng, Y.Zhou, R.Zhou and Z.Wang, "Overlapping Community Detection by Local Community Expansion", J. Inf. Sci. Eng, pp: 1213-1232, 2015.

[37]. S. Maity, S K. Rath; "Extended Clique percolation method to detect overlapping community structure", International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp: 31-37. 2014. IEEE

[38]. Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, Vol. 466, 2010, pp. 761-764.

[39]. S. Lim, S. Ryu, S. Kwon, K. Jung, J-G. Lee, LinkSCAN*: Overlapping community detection using the link-space transformation, 2014 IEEE 30th International Conference on Data Engineering. pp: 292-303. 2014.

[40]. M. Li, J. Liu, A link clustering based memetic algorithm for overlapping community detection. Physica A, 2018

[41]. F. Huang, X. Li, S. Zhang, J. Zhang, J. Chen, Z. Zhai, “Overlapping community detection for multimedia social networks”, IEEE Transactions on multimedia. pp: 1881-1893. 2017

[42]. A. Lancichinetti, F. Radicchi, Ramasco and S.Fortunato, “Finding statistically significant communities in networks” PloS one, 2011, Public Library of Science

[43]. JJ. Whang, DF. Gleich, IS. Dhillon, “Overlapping community detection using seed set expansion”, Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp: 2099-2108. 2013

[44]. Y. Zhou, G. Sun, Y. Xing, R. Zhou, Z. Wang, “Local community detection algorithm based on minimal cluster”, Applied Computational Intelligence and Soft Computing, 2016. Hindawi

[45]. X. Chen, J. Li, “Overlapping Community Detection by Node-Weighting”, ICCDA 2018, DeKalb, IL, USA. © 2018 Association for Computing Machinery

[46]. S. Gregory, ‘‘Finding overlapping communities in networks by label propagation,’’ New J. Phys., vol. 12, no. 10, p. 103018, 2010

[47]. X. Zhu, Z. Ghahramani; “Learning from labeled and unlabeled data with label propagation”, 2002.

[48]. Z.H Wu, Y.F Lin, S. Gregory, H.Y Wan, S. F. Tian; “Balanced multi-label propagation for overlapping community detection in social networks”, Journal of Computer Science and Technology, pp: 468–479, 2012.

[49]. J. Xie, B K. Szymanski, X. Liu, Slpa: Uncovering overlapping communities in social networks via a speaker listener interaction dynamic process. IEEE, pp: 344-349. 2011

[50]. M. Lu, Z. Zhang, Z. Qu, Y. Kang, “LPANNI: Overlapping community detection using label propagation in large-scale complex networks”, IEEE Transactions on Knowledge and Data Engineering 2018

[51]. A. Zakrzewska, D.A. Bader; “Tracking local communities in streaming graphs with a dynamic algorithm”, pp:16-Jan, 2016 Springer

[52]. N. Aston, J. Hertzler, W. Hu; “Overlapping community detection in dynamic networks”, Journal of Software Engineering and Applications, 2014

[53]. A. Mahfoudh, H. Zardi, M A. Haddar, “Detection of dynamic and overlapping communities in social networks”, Int. J. Appl. Eng. Res. pp: 9109-9122. 2018

[54]. S. Boudebza, R. Cazabet, F. Azouaou, O Nouali, “OLCPM: An online framework for detecting overlapping communities in dynamic social networks”, pp: 36-51, 2018, Elsevier

[55]. F. Ding, Z. Luo, J. Shi, and X. Fang, “Overlapping community detection by kernel-based fuzzy affinity propagation”, In Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on, pages 1–4, May 2010.

[56]. M. Magdon-Ismail and J. Purnell; “Fast overlapping clustering of networks using sampled spectral distance embedding and Gmms”, In SocialCom/PASSAT, pages 756–759. IEEE, 2011.

[57]. S. Zhang, R.-S. Wang, and X.-S. Zhang; “Uncovering fuzzy community structure in complex networks”, Phys. Rev. E, 76:046103, Oct 2007.

[58]. M. Zarei, D. Izadi, and K. A. Samani; “Detecting overlapping community structure of networks based on vertex–vertex correlations”, Journal of Statistical Mechanics: Theory and Experiment, 2009(11):P11013, 2009

[59]. J. Yang and J. Leskovec; “Overlapping community detection at scale: a nonnegative matrix factorization approach”, In WSDM, pages 587–596, 2013

[60]. A. McDaid, N. Hurley; “Detecting highly overlapping communities with model-based overlapping seed expansion”, international conference on advances in social networks analysis and mining, pp:112-119, 2010, IEEE

[61]. Y. Xiaobo; C. Chuxiang; W. Zhiwan; "Improved LFM algorithm in weighted network based on rand walk”, 2017 29th Chinese Control and Decision Conference (CCDC), 3719-3723 2017, IEEE

[62]. F. Havemann, M. Heinz, A. Struck, J. Gläser; “Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels”, Journal of Statistical Mechanics: Theory and Experiment, pp: P01023, 2011, IOP Publishing

[63]. M. Coscia, G.Rossetti, F. Giannotti, D. Pedreschi, “Demon: a local-first discovery method for overlapping communities”, Proceedings of the 18th ACM SIGKDD

international conference on Knowledge discovery and data mining, pp: 615-623, 2012

[64]. F. Reid; A. McDaid; N. Hurley; "Percolation computation in complex networks", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 274-281, 2012.

[65]. F. Moradi, T. Olovsson, P. Tsigas; "A local seed selection algorithm for overlapping community detection", 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), 2014, IEEE

[66]. R.V. Belfin; "Parallel seed selection method for overlapping community detection in social network", Scalable Computing: Practice and Experience, pp: 375-385, 2018

[67]. R.V. Belfin, P. Bródka; "Overlapping community detection using superior seed set selection in social networks", Computers & Electrical Engineering, pp: 1074-1083, 2018, Elsevier

[68]. H. Liu, L. Fen, J. Jian and L.Chen, "Overlapping community discovery algorithm based on hierarchical agglomerative clustering", International Journal of Pattern Recognition and Artificial Intelligence, 2018, World Scientific

[69]. K. Asmi, D. Lotfi; M.El Marraki; "A new local algorithm for overlapping community detection based on clustering coefficient and common neighbor similarity", Proceedings of the ArabWIC 6th Annual International Conference Research Track, pp: 6-Jan, 2019

[70]. J. Ma, J. Fan; "Local optimization for clique-based overlapping community detection in complex networks", IEEE Access, vol: 8, pp: 5091-5103, 2019, IEEE

[71]. A. Choumane, A. Awada, A. Harkous; "Core expansion: a new community detection algorithm based on neighborhood overlap", Social Network Analysis and Mining, vol: 10, pp: 11-Jan, 2020, Springer

[72]. K. Guo, L. He, Y. Chen, W. Guo, J. Zheng; "A local community detection algorithm based on internal force between nodes", Applied Intelligence, vol: 50, pp: 328-340, 2020, Springer

[73]. I. B. EI. Kouni, W. Karoui, L.B. Romdhane and Lotfi, "Node importance based label propagation algorithm for overlapping detection in networks", *Expert Systems with Applications*, vol. 162, 2020, Elsevier

[74]. K. Berahmand, A. Bouyer, M.Vasighi; “Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes”, IEEE Transactions on Computational Social Systems, vol: 5, pp: 1021-1033, 2018, IEEE

[75]. F. Cheng, C. Wang, X. Zhang, Y. Yang; “A local-neighborhood information based overlapping community detection algorithm for large-scale complex networks”, IEEE/ACM Transactions on Networking, vol:29, pp: 543-556, 2020, IEEE

[76]. K. Asmi, D. Lotfi, A. Abarda; “The greedy coupled-seeds expansion method for the overlapping community detection in social networks”, Computing, vol: 104, pp: 295-313, 2022, Springer

[77]. Y. Li, J. He, Y. Wu, R. Lv; “Overlapping community discovery method based on two expansions of seeds”, Symmetry, vol: 13, 2020, MDPI

[78]. J. Reichardt, S. Bornholdt, “Detecting fuzzy community structures in complex networks with a Potts model”, Phys. Rev. Lett. 93, 218701 (2004)

[79]. M. Rosvall, C. T. Bergstrom, “Maps of random walks on complex networks reveal community structure”, Proc. Natl. Acad. Sci. USA 105, 1118–1123 (2008)

[80]. H.W. Shen; “Community structure of complex networks”, 2013, Springer Science & Business Media

[81]. Z. Gao; “Community detection in graphs”, 2020, Indiana University

[82]. Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu and X. Yu; “Pathselclus: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks”. In: ACM Transactions on Knowledge Discovery from Data (TKDD) 7.3 (2013), p. 11.

[83]. A. Mirshahvalad; “Significant communities in large sparse networks”. In: PloS one 7.3 (2012).

[84]. Kevin S Xu and Alfred O Hero, “Dynamic stochastic blockmodels for time-evolving social networks”, In: IEEE Journal of Selected Topics in Signal Processing 8.4 (2014), pp. 552– 562.

[85]. Kevin Xu. “Stochastic block transition models for dynamic networks”. In: Artificial Intelligence and Statistics. 2015, pp. 1079–1087

[86]. D. A Spielman and S. H Teng, “A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning”, In: SIAM Journal on computing 42.1 (2013), pp. 1–26.

[87]. J. J. Whang, X. Sui, and I. S. Dhillon. “Scalable and memory-efficient clustering of large-scale social networks”. In: 2012 IEEE 12th International Conference on Data Mining. IEEE. 2012, pp. 705–714.

[88]. S. Papadopoulos, Y. Kompatsiaris and A. Vakali; "Community detection in social media." Data Mining and Knowledge Discovery 24.3 (2012): 515-554

[89]. F. D. Malliaros and V. Michalis; "Clustering and community detection in directed networks: A survey." Physics Reports 533.4 (2013): 95-142.

[90]. G. Rossetti, R. Cazabet; “Community discovery in dynamic networks: a survey”, ACM computing surveys (CSUR), 51 2, Jan-37, 2018 ACM New York, NY, USAP.

[91]. A. Karataş, S. Şahin; “Application areas of community detection: A review”,2018 International congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT),65-70, 2018, IEEE

[92]. H. Sarvari, E. Abozinadah, A. Mbaziira, and D. Mccoy, "Constructing and analyzing criminal networks," in 2014 IEEE Security and Privacy Workshops, 2014, pp. 84-91.

[93]. A. Karataş, and S. Şahin, “A Review on Social Bot Detection Techniques and Research Directions,” in Proc. Int. Security and Cryptology Conference Turkey, 2017, pp.156-161.

[94]. F. Taya, J. de Souza, N. V. Thakor, and A. Bezerianos, “Comparison method for community detection on brain networks from neuroimaging data,” Appl. Network Sci., vol. 1, no. 1, pp. 8, 2016.

[95]. D. Lalwani, D. V. Somayajulu, and P. R. Krishna, "A community driven social recommendation system," in Proc. 2015 IEEE Int. Conf. on Big Data, 2015, pp. 821-826.

[96]. A. Clauset, Finding local community structure in networks, Physical review E, vol 72, pg: 26132, 2005, APS

[97]. F. Luo, James Z. Wang, E. Promislow; “Exploring local community structures in large networks”, Web Intelligence and Agent Systems: An International Journal, vol 6, pp: 387-400, 2008, IOS Press

[98]. A. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah ; ”A comparison study on similarity and dissimilarity measures in clustering continuous data”, PloS one, 2015, Public Library of Science San Francisco, CA USA.

[99]. M. Chen, B K Szymanski; “Fuzzy overlapping community quality metrics Social Network Analysis and Mining”, vol 5, pp: 14-Jan, 2015, Springer

[100]. H. Wu, L. Gao, J. Dong, X. Yang; “Detecting overlapping protein complexes by rough-fuzzy clustering in protein-protein interaction networks”, PloS one, vol 9 pp: e91856, 2014, Public Library of Science San Francisco, USA

[101]. W. W. Zachary; ”An information flow model for conflict and fission in small groups”, Journal of anthropological research, vol 33 , pp: 452-473, 1977, University of New Mexico

[102]. D. Lusseau; “Evidence for social role in a dolphin social network”, Evolutionary ecology, vol 21, pp: 357-366, 2007, Springer

[103]. V. Krebs, unpublished, http://www.orgnet.com, 2019

[104]. A. Epasto, S. Lattanzi, R. Paes Leme; “Ego-splitting framework: From non-overlapping to overlapping clusters”, Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp: 145-154, 2017.

[105]. M. EJ. Newman; “Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality”, Physical review E, vol 6416132, 2001, APS

[106]. Steinhaeuser K, Chawla NV (2010) Identifying and evaluating community structure in complex networks. Pattern Recognition Letters 31: 413–421.