Community Detection in Complex Networks using a novel nature inspired algorithmic approach based on Ant Lion Optimizer

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Abhay Mahajan

Roll No. 601303001

Under the supervision of:

Dr. Maninder Kaur

Assistant Professor, CSE Department

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, *"Community Detection in Complex Networks using a novel nature-inspired algorithmic approach based on Ant Lion Optimizer"*, in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Maninder Kaur* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Abhay Mahajan)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Maninder Kaur)

Assistant Professor, CSE Department

Countersigned by

(Dr. Deepak Garg)

Head

Computer Science and Engineering Department

Thapar University

Patiala

(Dr. S. S. Bhatia)

Dean (Academic Affairs)

Thapar University

Patiala

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. Maninder Kaur**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions.

I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia,** Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother, since he insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: July, 2015
Place: Thapar University, Patiala

(Abhay Mahajan)

ABSTRACT

Most of the existing system in the world network is represented with the help of links and nodes in which nodes represent the systems and the links represent the relationship between the connecting or interrelating nodes. Some of the well-known existing networks are social media and online social networking websites like Facebook, Google+, and Twitter. The network links in different type of domains represent different types of relationships. E.g. human friendship, animal's physical proximity, interconnectivity of infrastructures, organizational structures, Web hyperlinks and abstract relationships like similarity between data points. The existence of communities shows the structure of the networks existing in nature. Communities, which are also named as modules or clusters, are the groups of relatively connected nodes, and are said to be intrinsic structures of the networks existing in nature. Nodes of the same community or cluster usually share common interesting properties such as a function, purpose and interest. Hence, one of the most crucial problems in network analysis is community detection.

Several methods have been proposed which help in detecting communities of the complex networks. Amongst them, the most popular technique is dependent on the optimization of the objective, modularity which is the most widely used function to evaluate the quality parameters of the group structure of networks. Various heuristic algorithms dependent on modularity optimization in detecting communities have been proposed in past few years. The problem of community detection in complex networks has been received an increased amount of interest since the past decade. Community detection is a way to uncover the structure of networks. This is done by grouping the nodes into communities. The grouping is done for the communities in which the interconnection between the nodes is found to be denser than the intra-connection between the communities. In this thesis, a novel nature-inspired algorithmic approach based on Ant Lion Optimizer is proposed which helps in discovering the communities efficiently in large networks. The proposed algorithm is termed as ALOCD for short. The algorithm optimizes modularity function and is able to identify densely connected groups of nodes having sparse interconnections. Experiments on real-world networks show the capability of the method to efficiently detect the network structure. In this, ALOCD algorithm is also compared with Ant Colony Optimization (ACO) algorithm and Enhanced Firefly Algorithm (EFF) for community detection.

CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

Real graphs are undoubtedly not homogeneous and largely differ from uniform random graphs. Nodes with very high and very low degree coexist in real networks. Vertices do not only differ globally in their degree distribution but also locally in their edges distribution. It has been observed that the local concentration of edges in specific group of nodes is much higher than the global density of the network. Most of the existing system in the world network is represented with the help of links and nodes, in which nodes represent the systems and the links represent the relationship between the connecting or interrelating nodes. Some of the well-known existing networks are social media and online social networking websites like Facebook, Google+, and Twitter [28].

The network links in different type of domains represent different types of relationships. E.g. human friendship, animals physical proximity, interconnectivity of infrastructures, organizational structures, Web hyperlinks and abstract relationships like similarity between data points [1]. The existence of communities shows the structure of the networks existing in nature. Communities, which are also named as modules or clusters, are the groups of relatively connected nodes, and are said to be intrinsic structures of the networks existing in nature. These groups of nodes found in real networks are known as community structures. An example of community structure is illustrated in Fig1.1. Nodes of the same community or cluster usually share common interesting properties such as a function, purpose and interest.

1.1. Community Structure

Community structure is the division of network nodes into groups within which the network connections are dense. Communities play a crucial role in understanding a network. As shown in Fig1.1 communities forms a cohesive group of highly interconnected nodes and the nodes within these dense clusters are expected to share many common properties or to fill a similar purpose in the graph. Furthermore, extracting communities can also reveal some behavioral or structural properties of the nodes. In social and metabolic networks, a core structure is often encountered within the different communities [10]. The core nodes provide

the stability of the clusters and are the most internally connected vertices. If such nodes are removed or inhibited, the community ceases to exist as a cohesive group. On the other hand, the peripheral nodes act as mediators between the different communities and allow the exchange of information.

This fundamental distinction between the roles of the nodes in a community has been particularly studied in epidemiology. It has been shown that the community structures may largely decrease the spread of infectious diseases because only the peripheral nodes can disseminate the infection between clusters. Furthermore, it is crucial to identify the core nodes as the target individuals for vaccination policies.

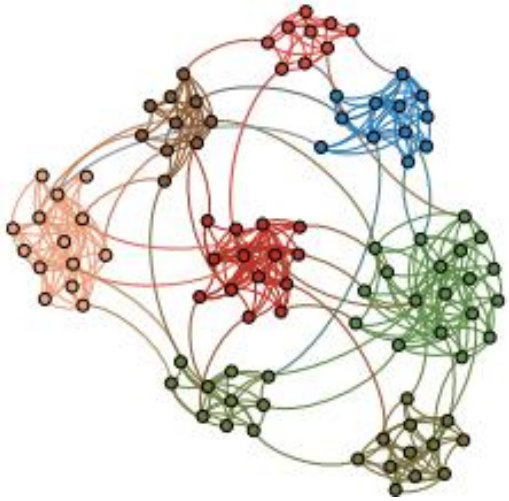

Fig1.1: Communities in Networks [28].

1.2. Applications of finding Communities

Communities arise naturally in most, if not all, kinds of social networks where people form spontaneously groups sharing common characteristics depending on different types of interactions like family, friends, work, sports and so on [1]. The extraction of relevant clusters in large social networks is a real challenge but provides accurate insights about the influence of individuals and spread of information. There are various applications [28] of finding communities:

- Communities can reveal unexpected behaviors. For example, Belgium is known to be mainly separated in 2 distinct regions using different languages, either French or Dutch.

- The extraction of communities in a mobile telecommunication network has highlighted the essential role of the capital, Brussels, for the communication between the 2 large communities. Surprising results have also been obtained in the analysis of mobile telecommunication networks of France and Great-Britain where the geographical borders of the departments can be recovered as the boundary of communities, even though the creation of those geographical borders goes back to several decades and should not influence how people interact nowadays.

- Community detection is also useful in other types of social interactions. Institutional and geographical constraints have proven to control the patterns of scientific collaborations, criminal networks have been studied to reveal the structure of large organizations of offenders and a network of bottlenose dolphins has been investigated to uncover their social strata.

- The potential applications of community detection go far beyond the analysis of social behaviors. Scientists have given evidence, using a goods exchange network, that trading communities reduce the probability of international conflicts. The worldwide air transportation network has been investigated to explain some anomalies in the airports connectivity, i.e. the most connected cities are not the most central ones, which reveal that geopolitical considerations have to be taken into account to understand the airports connectivity.

- Community detection has also yielded functional cartographic representations of the different roles of nodes in metabolic networks, known to be extremely difficult to study. It has been used to better understand the dispersal of coral larvae in the Great Barrier Reef which is essential to prescribe better management and control policies of the coral reef water.

- Community extraction has also been applied in image processing to reveal the contour of objects and to track moving bodies in video.

1.3. Community structure in graph partitioning and sociology

Community structure in networks is closely related to graph partitioning in graph theory [5] and computer science, and hierarchical clustering in sociology [21].

1.3.1. Graph Partitioning

Graph partitioning is to identifying coherent groups of nodes that are on average more similar one with each other than with the rest of the network. Therefore, graph partitioning is always tied to a fitness measure or a cost function to optimize. The elementary formulation of the partitioning problem is to define a cut, i.e. a subset of the edges such that removing those edges creates 2 disconnected components in the graph [5]. The objective function discriminates the quality of different cuts. The first type of cut is the minimum cut and consists in identifying the minimal set of edges that bisects the graph. The minimum cut problem is equivalent to the maximum flow problem, i.e. by identifying the minimum cut, one can compute the maximum possible flow that can be sent through the network. Unfortunately, the minimum cut criterion tends to produce an unbalanced partition, i.e. for unweighted network, the optimal solution consists in identifying the node of minimum degree and to select its incident edges as the cut set [24].

Other types of cuts have been defined like the maximal cut, which consists in finding the maximal set of edges that bisects the graph, or the sparsest cut which consist in identifying the cut that minimizes the ratio between the number of edges in the cut and the number of nodes in the smallest of the disconnected components. However, unlike the minimum cut which can be solved in polynomial time, many cut problems like the maximum cut and the sparsest cut are NP-hard. One possible way to define such an approximation is to use spectral partitioning which consists in defining a cut based on the eigenvectors and eigenvalues of an appropriate matrix.

Another approach to define graph partitioning is to somehow measure how central are each node or each edge in the network. It has been observed that most of the edges with large weight are often lying in the middle of densely connected clusters of nodes and that it is in fact the edges with small weight that maintain the global connectivity of the network and allow the exchange of information between clusters. Following this principle, different measures of centrality have been defined like the edge betweenness centrality [26] which is, for each edge, the number of pairs of nodes (i, j) such that at least one of the shortest path between them contains the edge. The weak ties are the edges that should define the cut and have a high betweenness centrality since they will often be involved in the shortest paths between nodes of different clusters. Therefore partition can be defined by sequentially

removing the edge with the higher betweenness centrality until an appropriate number of clusters have been found.

1.3.2. Sociology of communities

In sociology, the terms groups and communities are often used interchangeably to refer to the same concept of cohesive groups [28]. Sometimes, the term subgroup is used instead if the term group is used to refer to a sample of the population. All sociological groups or communities have the following properties [15] in common:

- **Cohesiveness:** Cohesiveness is the force that holds a group together. The level of cohesiveness varies according to many factors such as existence of threats from the outside, strength of leadership, group identity and objectives, and relationships among group members.

- **Norms:** Group norms are the standards of behaviors which arise naturally until they have become tradition-like. Group norms are the laws of the group and reveal things that a group holds as values.

- **Sanctions**: Group sanctions are the policing force which enforces the group norms. Norms of a group change over time. A new norm maybe introduced to the group, while an old norm vanishes unless the group members maintain it.

- **Objectives**: Group objectives are the reasons why a group exists. The objectives of the group come from all of its members who continue to maintain those objectives. Objectives may change from time to time according to internal and external changes, so that the group remains competent.

- **Self-perpetuation**: A group is a living entity which has a desire to survive and continue to live. The most important element to self-perpetuation is the leader-follower relationship. The health and survival of the group depend on decisions made by the leader and how the followers evaluate those decisions as fitting the group objectives.

All groups share the same set of properties to some extent. For a group to exist and continue to survive there must be some cohesive force which holds its members together.

1.4. Community Detection in Networks

Communities are areas in the network which have relatively more connections within the areas than across different areas [28]. However, no single formal definition of communities is universally accepted. Even then, in many cases, researchers usually start by stating some intuitions about communities and then proceed to describe a heuristic algorithm which finds communities that capture the intuitions, without formally define what communities are. Such notions of communities are usually termed as being operationally or algorithmically defined, which essentially means that communities are whatever the heuristic algorithms output.

1.4.1. Local Notions

In these notions, communities are viewed as independent patterns which repeat themselves and the main concerned is finding all community-like patterns without worrying about how the patterns fit together into one picture of overall community structure [36]. To some extent, such communities can be considered as independent units. This was how the notion of communities was viewed in the early development of social network analysis, which was heavily influenced by graph theory. As a result, the early definitions of communities are mostly graph-theoretic and local.

The simplest, yet strictest definition of a community is a clique, which is a complete subgraph. Every vertex is adjacent to every other vertex in a clique. Because of its restrictive nature, many researchers have proposed better alternatives of cliques by considering the diameter of a (sub-) graph. Cliques are subgraphs with diameter one so other subgraphs with small diameters can potentially be community-like as well [33]. This gives three variations of cliques: n-cliques, n-clans, and n-clubs. All are maximal subgraphs of diameter n. The subtle differences between n-cliques, n-clans, and n-clubs are in the choices of graphs and subgraphs used in computing the distances. n-cliques use the distances in the original graph, n-clubs use the distances in the subgraphs and n-clans use the distances in both the original graph and the subgraphs.

These choices affect the computational complexity of the algorithms for finding them. Two cliques of size k are considered adjacent if they share all but one vertex. If two cliques overlap one another significantly, then they would be put in the same community [12]. This

notion of clique adjacency is used to define a notion of community, called clique percolation, which is the maximal collection of adjacent cliques of the same size.

1.4.1.1. Cohesive or Fitness measures of Local Notions

Cohesiveness measures are also known as fitness measures [18]. The higher the measure on a subgraph, the more community-like the subgraph is.

- **Intra-cluster density measure:** The first is intra-cluster density which is the fraction of the number of edges in the subgraph.
- **Relative density measure:** Another measure is relative density which is the ratio between the number of intra-cluster edges and the number of edges which touch the subgraph.
- **Conductance measure:** Another measure of how disconnected a subgraph is from the rest of the network is conductance, which is the ratio between the size of the cut and the volume of the set. The lower the conductance value, the more community-like a candidate set is. Thus, a community is defined as a candidate set whose conductance is no greater than a certain threshold.

The problem with using local definitions is that they treat communities as independent entities out of context. Local approaches lack the capability to explain how communities in a network interact with one another.

1.4.2. Global Notions

Compared the local notions of communities, the global notions are based on the structural information about the whole graph. Local definitions are usually self-centric in nature. Global notions look at the network link structure and all candidate sets that form a community structure as a whole [36]. In particular, a good quality function should capture the notion of a good collection of communities. A quality function is a function which takes a partition of the vertex set of a graph and outputs a number which quantifies how likely the parts of the partition are communities of the graph. Since the notion of communities itself is not universally defined, therefore each quality function captures only one aspect of communities.

1.4.2.1. Quality Functions of Global Notions

There is no universally accepted definition of communities. Communities are defined as sets of nodes with a high internal density, either in the number of internal edges or their weight, and a low external density with the rest of the network. This implies that the subgraph induced by a community should be at least weakly connected; otherwise the total internal density could be increased by partitioning the community into its weakly connected components without increasing the external density [26]. The most straightforward description of communities is to assume that they are spanned by the densest possible clusters, i.e. maximal complete subgraphs or cliques [33]. Nevertheless, this characterization is quite restrictive and does not fit for real networks.

A large internal density is not the only defining property of a community which also requires a low external density to be accurately characterized. This leads to the definitions of strong and weak communities. A community is said to be strong if the internal degree of all its vertices is higher than its external degree. This definition is also quite restrictive and barely found in real networks. Hence, communities have also been defined in a weak sense as clusters with a total internal degree i.e. the sum of the internal degree of its vertices are larger than its total external degree. While those characterizations of communities are more appropriate for real networks, they lack the formalism to create an algorithm to actually extract community structure from a given network. Therefore, it is often more convenient to establish a global definition of communities by using fitness measures or quality functions [13] over the entire graph.

- **Performance quality function:-** This quality function is based on the fraction of all vertex pairs classified correctly by the partition.

- **Coverage: -** Another quality function is the coverage which is the fraction of edges of the graph which fall inside the parts of the partition.

Another way to define a global notion of communities is to use a null model. A community structure is an evidence of heterogeneity in a social network. Links are biased toward vertices which belong to the same communities. On the other hand, a random graph, imposes homogeneity of link structure, and thus is unlikely to have a community structure. This leads to the definition of modularity by Newman and Girvan.

1.4.3. Modularity by Newman and Girvan

Newman and Girvan have introduced a fitness measure called the modularity, based on the configuration model that produces a random graph with the same degree distribution as the original network. The modularity has rapidly become one of the most popular cost functions for community detection [25]. The modularity score of a partition ranges in [−1, 1]. Community partitions associated to high positive values of the modularity score are supposed to accurately represent the modular structure of the network. It was originally assumed by Newman & Girvan that modularity value larger than 0.3 for a given partition indicates that the network has a community structure accurately represented by the aforementioned partition. The modularity (Q) in case of modules [25] given by Newman and Girvan is defined as:

$$Q = \sum_{i=1}^{n} \left[\frac{l_i}{L} - \left(\frac{d_i}{2L} \right)^2 \right]$$

Where, l_i is the number of links b/w nodes in module S, L is the number of links in network, d_i is the Sum of degrees of the nodes in module S and n is the total number of modules.

Newman and Girvan further modified the modularity [26] as,

$$Q = 1/2m \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(i,j)$$

Where, $A_{i,j}$ is the adjacency matrix, m is the number of edges in network, d_i, d_j are the degree (or strength) of nodes i, j and $\delta(i,j)$ is the function which return 1 when both i, j are in same community else 0.

The intuition behind the formula is that it compares the number of intra-cluster edges in the original graph with the expected number of intra-cluster edges when the graph is randomly rewired based on the degrees of the nodes [21]. Modularity was proposed to be a way to automatically discover the true number of communities.

1.4.4. Different types of Clustering or Grouping Techniques

1.4.4.1. Data Clustering

Data clustering which is also known as cluster analysis, [17] differs from the community detection problem in that the entities being clustered are data points from some ambient space. In other words, data clustering tries to group items based on their attributes while community detection tries to group items based on their relationships. The notions of similarity and distance are commonly used in data clustering [16]. To use a data clustering technique to do community detection, one starts with embedding a social graph in some space so that data clustering methods can be used. For example, spectral graph clustering embeds the vertices in the eigenspace of some graph Laplacian [30]. It then derives some similarity measures from the graph topology and finally applies data clustering methods. Some data clustering techniques, such as hierarchical methods, do not require an embedding of graph in some space as long as the similarity between every pair of data points can be computed.

1.4.4.2. Distance based Clustering

Many community detection algorithms use transitional techniques developed for data clustering, also known as cluster analysis. In distance based clustering, data points belong to some vector space in which a norm or distance is defined. The number of expected clusters k is usually given as a parameter[16]. The most popular technique is perhaps k-means in which one wants to cluster the data points such that the intra-cluster sum of squared distances is minimized. The problem is NP-hard [24]. Other variations of k-means are k-centers, k-medians, and k-medoids. One advantage of these methods is that they not only partition the data points but also partition the ambient space. For example, the result of k-means is a partition of the data space into Voronoi cells [30]. Thus, the result can be used to classify new data points by assigning each new data point to the cluster whose mean is nearest.

1.4.4.3. Hierarchical Clustering

Hierarchical clustering is the set of techniques that aimed at discovering natural divisions of networks into groups [21]. Community structures in the real world usually have nested structure. Examples of different granularities of communities include townships, counties, states, countries. Thus, it is more reasonable to model a community structure as a dendrogram as depicted in Fig 1.2 instead of a partition. The hierarchical clustering method produces as an output a dendrogram, which formally is a chain of partitions in which each

partition is finer than the next. For two partitions P and Q of the same set, P is finer than Q if and only if every part of P is contained in some part of Q. A chain of partitions expresses the nested structure from the finest partition to the coarsest partition. Hierarchical methods require a similarity or dissimilarity function between the data points [26]. They do not need the data points to belong to some ambient space. This sometimes makes them very slow in practice since the similarity function might be computationally hard.

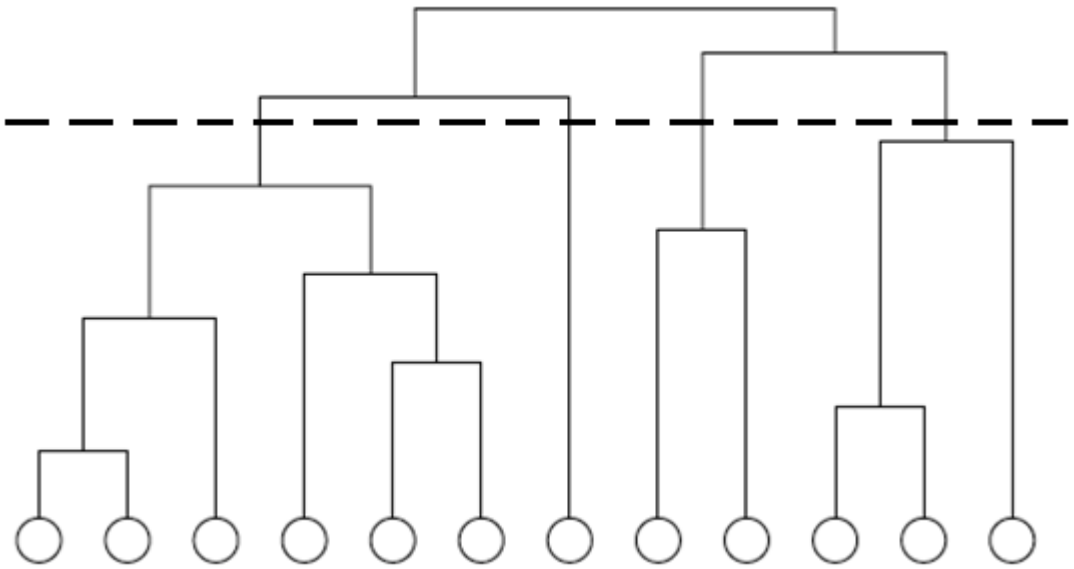

Fig1.2: Hierarchical Clustering [26].

Hierarchical clustering is divided into two methods i.e. agglomerative and divisive [21, 26].

1.4.4.3.1. Agglomerative Clustering

Agglomerative algorithms start from all vertices represented as different clusters as depicted in Fig. Then, two clusters are merged based on their similarity. The merging continues until only one cluster is left. There are several ways to define the similarity of two clusters, which gives rise to different agglomerative algorithms.

- **Single linkage** defines the similarity between two clusters as the maximum similarity or more commonly the minimum distance between the vertices from the two clusters.

- **Complete linkage** which is the opposite of single linkage, defines the cluster similarity as the minimum similarity or maximum distance between the vertices from the two clusters.

- **Average linkage** defines the cluster similarity as the average of all pairs of vertices from the two clusters. Average linkage is also known as UPGMA which stands for Unweighted Pair Group Method with Arithmetic Mean.

Algorithm

Given a similarity matrix.

a) Firstly, the merging of two most similar matrix is done.

b) Then, updation of the similarity matrix is done to show that the similarity between the new cluster and the existing clusters is included.

c) Repeat (a) and (b) until all clusters merged into one cluster.

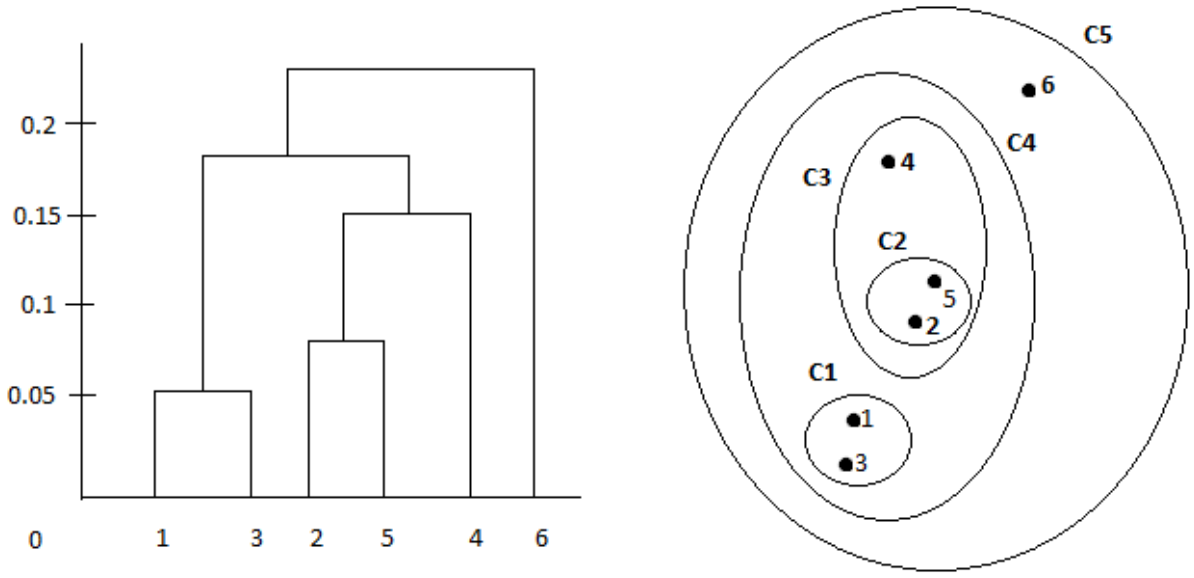

Fig 1.2.1: Agglomerative Clustering [3]

1.4.4.3.2. Divisive Clustering

Divisive algorithms start with all vertices in one big cluster. Then, a cluster is spliced into two based on the members' similarity. The process continues until every vertex is in a cluster by itself. An example is CONCOR which stands for Convergence of iterated

Correlations [25]. CONCOR is based on the observation that, if one repeatedly computes the correlation matrix from a correlation matrix, the entries of the result converge to either +1 or -1. Moreover, the rows and columns of this matrix can be simultaneously permuted so that the entries are grouped together into four blocks such that the two blocks along the main diagonal are all 1 and the two off-diagonal blocks are all -1. The block structure can then be used to bisection the data points.

1.4.4.4. Spectral Graph Clustering

Spectral clustering uses eigenvectors of matrices of similarity between objects to partition the data points. The idea is to change the basis of a matrix of similarity to the eigenspace of the matrix. Then, the eigenvectors are clustered using standard techniques such as k-means or a hierarchical method. This helps, for example, cope with the limitation of k-means that it produces a partition of space into convex Voronoi cells [30]. For spectral graph clustering, the adjacency matrix of the graph is transformed into a Laplacian matrix. Then, the first few eigenvectors of the Laplacian are computed .These eigenvectors are then used as a basis and the k-means clustering is performed on the data points which are projected on the eigenvectors [33]. An unnormalized Laplacian tends to favor clusters with intercluster density, while a normalized Laplacian tends to balance the high intra-cluster density and low inter cluster density.

1.4.4.5. Correlation Clustering

Correlation Clustering is the problem of clustering a graph with real-valued edge weights. Positive edge weights signify the confidence levels that the two end points should be clustered together, while negative edge weights signify the confidence levels that the two end points should not be clustered together. There are two versions of Correlation Clustering problem [9]. We can either maximize the sum of intra-cluster edge weights or minimize the sum of inter-cluster edge weights. The former is known as Maximizing Agreement and the latter is known as Minimizing Disagreement. The optimal solution of the two problems is the same, but approximation algorithms are different since one problem is maximization and the other is minimization. Maximizing agreement is NP-hard and APX-hard. Minimizing disagreement is NP-hard and APX-hard and has O (log n) approximation algorithms.

Approximation algorithms usually perform worse than heuristics on real-world data since they guarantee their performance in all cases, including rare cases. However,

approximation algorithms provide not only a solution to the problem but also a bound on the optimal solution.

1.4.5. Modularity Optimization Algorithms

Modularity is a popular quality function among network researchers. There are a few algorithms which aim to optimize this function in particular.

1.4.5.1. Divisive Algorithm based on Edge Betweenness

Girvan and Newman proposed one of the first successful algorithms to partition a graph into communities. Edge betweenness measures the number of all shortest paths passing a link, which gives the measure of the importance of that link when connecting two communities [26]. The most important link according to the edge betweenness i.e. the link with high edge betweenness as shown in Fig 1.3 is removed and the edge betweenness of the remaining edges is recalculated. This process is repeated until all links are removed. The recalculation of the edge betweenness after each removal is time consuming for finding the community structure for large networks but it also produces good results.

Algorithm

- Firstly the removal of the edge with the highest betweenness centrality is done.
- After that the edge betweenness for the remaining edges in the network is recalculated.
- Repeat all the steps until there are no remaining edges.

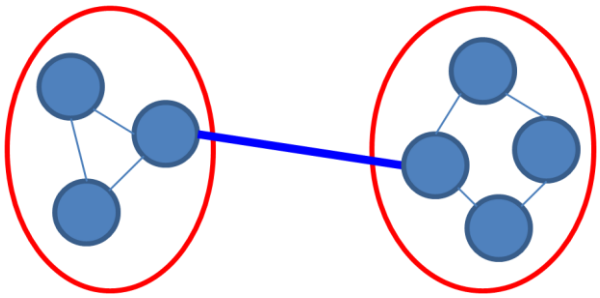

Fig 1.3: Two communities linked by brightest link with high betweenness [26]

1.4.5.2. Greedy Community Detection

After introducing the modularity as a quantitative criterion for the quality of a community partition, Newman proposed to optimize this measure using a greedy hierarchical agglomerative algorithm, although this algorithm can also be applied to optimize other additive cost functions [25]. Newman's algorithm starts with an initial partition with as many communities as nodes in the network, i.e. each node defines its own community. Then, communities are repeatedly joined in pairs. At each step, the join that provides that largest increase or the smaller decrease of the cost function is chosen among all possible joins [22]. Since the network contains initially n communities and at each step 2 communities are merged, reducing the number of communities by 1, the maximum number of joins is n – 1. This can be represented by a dendrogram, i.e. a tree showing the order of each join, as illustrated in Fig1.4.

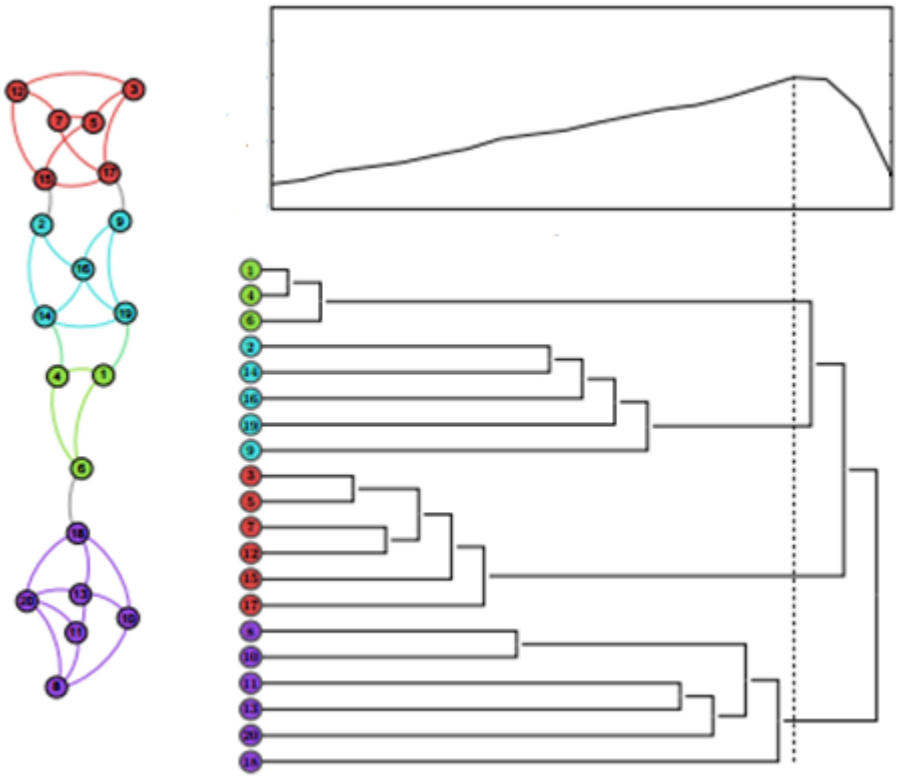

Fig1.4: Input graph and dendrogram created by Newman's algorithm [22]

A cut through this dendrogram defines a community partition for which one can compute the associated value of the fitness measure. Therefore, the best partition can be selected among all the levels of the dendrogram, which is represented by the dashed vertical line in Fig1.4. At each iteration, the algorithm must choose the best possible join. Yet

merging a pair of communities between which there are no edges can never improve the value of the cost function. Hence, the algorithm only considers pairs of communities that are connected by at least one edge, so in the worst case, it needs to check for O (m) possible joins.

1.4.5.3. Label Propagation

The label propagation algorithm extracts a stable partition for the resolution-limit-free model. In this framework, a partition is stable if there is no individual vertex switch from one community to another that increases the quality of the partition. The label propagation algorithm starts in a configuration where each node has a distinct label. Communities are defined during the whole process as sets of nodes with the same label [14]. Hence the algorithm is initialized with a partition in which each node defines its own community. Then, the labels propagate through the network using a simple updating rule i.e. a node is selected and its label is updated to the label shared by most of its neighbors. When there is a tie, one label is chosen uniformly at random. As the labels propagate, some of them will disappear while others will dominate inside groups of highly connected nodes.

The node selection is carried out by going through a randomly shuffled vector containing all the node indices. In one iteration, all the nodes have their labels updated at most once, then a new randomly shuffled index vector is created for the following iteration. The updating of the node labels can be performed either synchronously in which at iteration t, a node takes the label shared by most of its neighbors at time t – 1 or asynchronously in which at iteration t, a node takes the label shared by most of its neighbors at time t for the nodes that have been already updated, and at time t - 1 for the others. The algorithm stops when each node has a label shared by most of its neighbors.

Due to the random selection of labels when ties occur, it is often not possible to reach a stable labels configuration. Due to the randomness involved both in the nodes selection and in the tie breaking processes, multiple stable label configurations may be obtained from the same initial configuration. Raghavan et al. [34] showed that different community partitions obtained for a single network are in general similar. Though, one cannot choose easily a specific partition as the best one among all available partitions. Therefore, the proposed solutions are to either consider overlapping communities or to aggregate the different partitions to form a new overall partition.

1.4.5.4. Louvain Algorithm for Community Detection

The Louvain method [35] is a greedy hierarchical clustering algorithm. This algorithm has become extremely popular in recent years because it is fast, allowing to analyze huge networks with billions of edges, and produces significant partitions. For example, this algorithm is now implemented within the famous business social network LinkedIn and allows everyone to visualize the relationships between colleagues or customers and to better comprehend its professional network. Fig 1.5 shows the procedure of Louvain method and it is divided in two phases

- **Optimization phase:-** The first phase is the optimization phase which looks for a locally optimal partition by considering only individual vertex swaps. This phase of the algorithm is similar to the vertex mover refinement step of the algorithm of Schuetz and Caflisch [27] . As in many other greedy algorithms, the community partition is initialized with a single node per community and as many communities as nodes in the input network. Then, based on the modularity function, individual nodes are removed from their current community and swapped to the neighboring community which produces the largest positive gain of the modularity function. The nodes are potentially switched in a random order. Each node is considered once before a new random order of the nodes is created. When no more local correction with a non negative gain can be applied to increase the value of the modularity function, the algorithm initiates the second phase.

- **Aggregation phase:-** The second phase of the algorithm is the aggregation phase [35] in which the input graph is collapsed according to the communities found in the first phase. This creates a new network where each single node corresponds to a community as depicted in Fig 1.5. The purpose of this aggregation phase is to reapply the optimization phase on the collapsed graph such that one can consider additional improvements of the cost function by clustering groups of nodes rather than individual nodes.

The two phases of the Louvain method are recursively applied until no cluster increasing the value of the modularity function can be found in the last aggregated graph. In other words, the stopping criterion is that the optimal partition of the last aggregated graph. This algorithm produces hierarchical levels for the communities, i.e. the partition at a specific

scale is defined by merging communities of the partition at a lower scale. This algorithm has proven to produce good community structures. Its complexity is expected to be O (n log n) but a precise complexity analysis is still lacking due to the hardness to describe a priori the number of corrections through all the nodes in the optimization phase. Although the algorithm is fast in practice, the sequential corrections slow it down and make it very hard to parallelize on a multiple core architecture which in turn reduces the possible applications to very large networks.

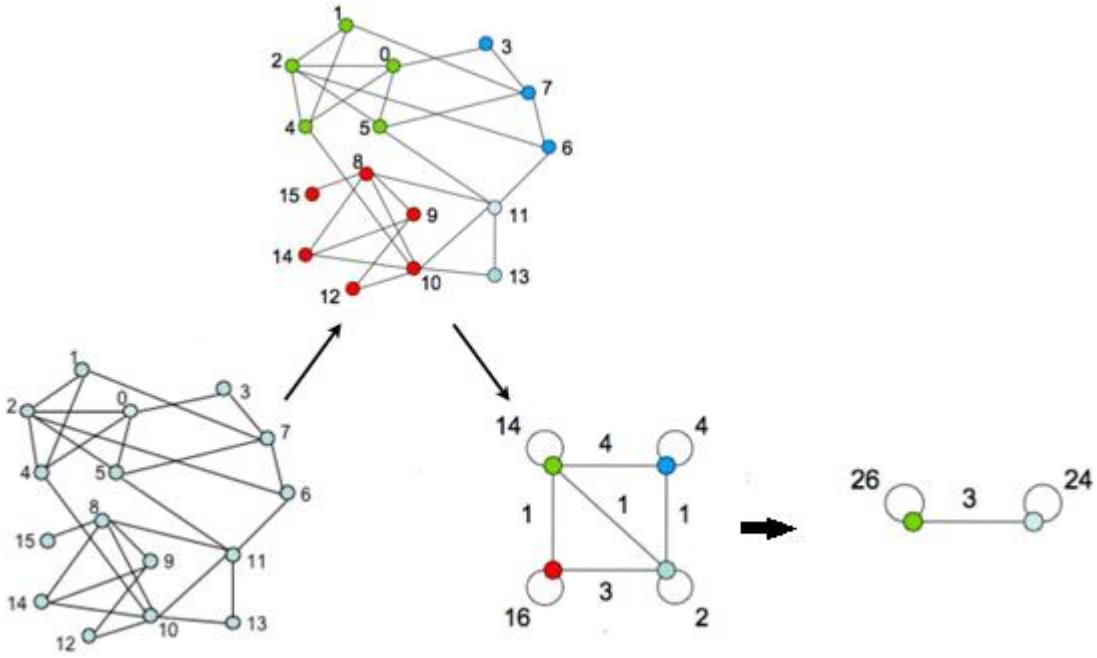

Fig1.5: Louvain Community Detection [35]

Chapter 2

Literature Survey

Various contributions have been anticipated by the researchers in the field of community detection in recent years to detect communities in complex networks, with each methodology being classified according to its algorithm type. This chapter categorizes the literature survey in the field of community detection based on analytical and evolutionary approaches.

2.1. Analytical Approaches

Kirkpetrick et al [31] proposed a discrete optimization algorithm named as Simulated annealing (SA). Simulated annealing (SA) is a general discrete optimization technique designed to allow, at the beginning, the exploration of a large proportion of the admissible set and to progressively narrow its search space. This algorithm has been applied for community detection by defining two types of transitions:

- The first type of accepted transition is a local vertex switch, i.e. a random node is taken out of its current community and assigned to another neighboring community.
- The second type of transition is more global and consists of either a merge of 2 communities or a split of a community into 2 distinct sets. The splits can be carried in different manners, for example via spectral bisectioning or constrained simulated annealing of the subgraph induced by the community.

It has been shown that applying global transitions leads to better approximations of the optimum than using only local moves. Typically, in practical applications, one considers n^2 local moves and n global moves before updating the temperature of the system. However, the complexity of the algorithm cannot be estimated and depends on the initial temperature and the cooling process. Though, the algorithm is typically very slow and should not be applied to graphs with more than 10^4 nodes.

Girvan and Newman [26] proposed one of the first successful betweenness based

divisive algorithms for community detection. The authors used the concept of edge-betweenness determining the edge importance in connecting the nodes via shortest paths. The algorithm first computes the betweenness of all edges and then removes the edge with high edge-betweenness, one at a time until the graph disconnected. The whole process is repeated until there are no more edges. The proposed approach determined the communities but it could not determine the strength of communities formed.

Newman et al. [22] proposed an algorithm based on agglomerative clustering in which they used the modularity function that determines the strength of communities and thus helped in overcoming the drawback of edge-betweeness based divisive algorithm. Their algorithm starts with all nodes in communities by themselves, after that it find two communities for merging such that the modularity value increases the most. Merging of two communities continued till all nodes are in the same community. This algorithm was efficient in case of speed but in practice, the modularity produced by this algorithm was not always high.

Clauset et al. [2] observed that the Newman's approach was not efficient for a sparse network as well as it was also inefficient with respect to time and memory. They did not describe a new algorithm but rather improved the implementation of Newman's original algorithm by using much more efficient data structures called max-heap. A heap is a tree-based data structure with a key for each node in the tree and an order relation on those keys. A heap is a max-heap if the key of a parent node is always greater than the keys of its children. Obviously, the largest key in a max-heap is always the root node but note that there is no specific ordering between siblings or cousins in the tree. The algorithm can be summarized as follows:

1. Compute the initial matrix M and store it as a balanced binary maxheap for each row. Compute the elements of the vector K for each node. Populate the max-heap L with the largest element on each row of M. Set H to its initial value.

2. Select the largest element of M from L, join the corresponding communities and update M, L and K. Update H = H+M (i, j).

3. Repeat step 2 until the largest element of L is negative. The value of H is the approximation of the maximum value of the cost function.

Their algorithm complexity is O (n $\log^2$n) for sparse networks which was the first algorithm used for analyzing large networks (about 10^6 nodes). The algorithm has a drawback that it might form large communities in the early phase at the cost of existing small communities.

Newman [23], proposed a new method for optimizing the fitness function with the help of associated matrix eigenvectors and eigenvalues called as Spectral optimization. Spectral optimization refers to the optimization of the fitness measure H using the eigenvectors and eigenvalues of the associated matrix and is similar to the classical spectral partitioning.

$$H = \sum_{i,j \in V} B(i,j)\delta(i,j)$$

Where the matrix B depends on the chosen cost function. This algorithm has been shown to work very well for graph bisection, however its performance declines when the graph contains more than 2 clusters, i.e. in the early stages of the algorithm, spectral partitioning tends to cut some of the existing clusters which become impossible to recover afterwards. Additionally, if some elements of the dominant eigen vector v1 are close to 0, then the assignments of the connected nodes may be ambiguous. The cut value of 0 that defines the partitioning vectors is somehow arbitrary and a slight variation of this threshold can lead to very different community partitions.

Some authors have proposed to use more than a single eigenvector to define the partitioning vector. In this case, the p dominant eigenvectors are computed and form an indicator matrix $X \in IR^{nXp}$, i.e. each row of X is an indicator vector of dimension p for the corresponding node. The community partition of the graph is then obtained by clustering the nodes in the induced p-dimensional space. Unfortunately, one needs to find an appropriate value for p, neither too large to reduce the computational complexity, nor too small to have sufficient information in the indicator vector to achieve good performances.

Schuetz and Caflisch [27] observed the bias of the algorithm of Clauset et al. towards the formation of large communities during the early stages. They proposed an alteration of the algorithm of Clauset et al. to promote the creation of multiple clusters at the same time, hence reducing the risk of condensation of the nodes into a few large communities. This task is carried out by allowing at most l independent joins in each level of the dendrogram. The qualifier "independent" for the joins refers to a "Touched-Community-

Exclusion-Rule" (TCER): a join between 2 communities is accepted only if none of the 2 communities have been involved in a previous join at the current dendrogram level. Each level of the algorithm can be summarized as follows:

1. Initialize: mark all the communities as "not touched".
2. Select the best join $\Delta H\ (i \cup j)$ over all remaining connected pairs of communities (i, j) where neither i nor j is marked as "touched".
3. If the gain $\Delta H\ (i \cup j) > 0$, apply the join and tag both communities I and j as touched. Otherwise, discard the pair (i, j).
4. Repeat steps 2 and 3 until l joins have been applied, or the best gain is negative $\Delta H\ (i \cup j) < 0$, or all the communities are tagged as touched.

The algorithm had same complexity as clauset et al. algorithm but instead of creation and maintenance of the max-heap, it required the computation of pairwise gains that increases the computational cost.

Blondel et al. [35] proposed a greedy hierarchical clustering algorithm named as Louvain method. The objective function considered in this algorithm was modularity. The algorithm is divided into two phases, with one as optimization phase and other one as aggregation phase. In optimization phase, the individual nodes are treated as individual communities and on the basis of modularity optimal partitioning is done i.e. the individual nodes which are considered as individual communities are removed from their current and placed into their neighbor communities with highest modularity value. This phase is considered till there is no community with non-negative value. In aggregation phase, the communities formed after optimization phase is considered as input i.e. the communities formed after optimization is considered as individual communities and again the optimization is applied on these communities thus by improving the modularity function by making the community of group of nodes instead of individual nodes.

These two phases of Louvain are functional until there is no positive change in modularity found in last aggregated network. The complexity of this algorithm is O (n log n). This algorithm is fast but on multiple core architecture, the sequential corrections makes it slow and it is also inefficient to be applied for very large networks.

Erwan Le Martelot and Chris Hankin [11, 12] proposed a new method which is based on two criteria one is global and other is local. Therefore authors proposed two new algorithm based on these two criteria:

- **Global criteria algorithm:** This algorithm is almost similar to Louvain method but it has one advantage over latter that it can be applied for multiscale detection of communities. Like Louvain this algorithm also have two phases. In first phase, the individual nodes are treated as individual communities and on the basis of modularity grouping of one with its respective neighbor community is done. In second merging phase, the communities formed in first phase are taken as input and merging of those communities further takes place in order to improving the modularity value.

- **Local criteria algorithm:** Local criteria algorithm is used in overlapping communities. The first phase is similar to global criteria but the second phase is not compulsory in this criteria until the overlapping of communities is more than half i.e. merging of communities is done only if communities significantly overlap.

The complexity of global criteria algorithm is O (n) whereas in local, the complexity is O (n^2) but local criteria has its advantage that it can be used for overlapping communities.

Table 2.1 shows the summarization of various analytical approaches:

Authors	**Year**	**Proposed Method**	**Fitness / Objective Function**	**Community Detection**	
				For Overlapping / Non-Overlapping Communities	**In Static / Dynamic Networks**
M. E. J Newman and M. Girvan	2003	Edge-Betweenness Based Divisive Algorithm	Edge-Betweenness and Modularity	Non-Overlapping	Static
Blondel et al.	2008	Louvain Method	Modularity	Non-Overlapping	Static

Huang et al.	2011	Local Tightness Expansion (LTE) Algorithm	Tightness Function Based on Similarity Measure	Overlapping and Non-overlapping Both	Static
E. L. Martelot and C. Hankin	2013	Global and Local Criteria	Modularity, LFM, LTE and NMI	Overlapping and Non-Overlapping Both	Static
E. L. Martelot and C. Hankin	2014	Local Criterion within a Multi threaded Algorithm	LFK Criterion and NMI	Overlapping	Static

2.2. Evolutionary Approaches

Clara Pizzuti [7] presented a genetic algorithm for uncovering communities in large networks named GA-Net. They introduced the concept named community score. The underlying objective of the algorithm is to maximize the value of community score for obtaining optimal partitioning of the network. The author also introduced the Safe individual criteria in genetic algorithm to avoid the useless computation thus by making the algorithm efficient. This algorithm had been applied on an undirected graph and the adjacency matrix is assumed as symmetric. The community score (CS) which had been introduced by author is defined as:

$$CS = \sum_{s=1}^{k} Q(V_s)$$

Where, k is the no of partitioning of a sub-matrix V i.e. $\{V_1 \dots \dots V_k\}$ and Q(V) is the score of sub-matrix V which is defined as $Q(V) = M(V) \times v_V$

M (V) is the power mean of V = (I,J) of order t and defined as $M(V) = \frac{\sum_{i \epsilon I} (a_{iJ})^t}{|I|}$

v_V is the volume of sub-matrix V = (I,J) is the number of entries of a_{ij} which are 1 and is defined as $v_V = \sum_{i \epsilon I, j \epsilon J} a_{ij}$

a_{iJ} is the mean value of ith row of the V and is defined as $a_{iJ} = \frac{1}{|j|} \sum_{j \epsilon J} a_{ij}$

The author had tested the algorithm for synthetic as well as real world data set and found comparable results which showed that this algorithm is effective in finding the communities in networks.

Clara Pizzuti [8] further modified his work by presenting a new genetic algorithm to discover optimal communities in complex networks named MOGA-Net i.e. multiobjective genetic algorithm basically a Nondominated Sorting Genetic Algorithm (NSGA-II) that builds and ranks the competing individuals population on nondominance basis. The approach uses hierarchal clustering technique producing network communities at different level of hierarchies and automatically resolute communities formed. For partitioning of network, the algorithm used modularity criteria and for measuring the performance of algorithm Normalized Mutual Information (NMI) had been used. The author had also tested the algorithm for synthetic as well as real world data set and found effective results.

Amiri et al. [4] proposed a new multi-objective optimization algorithm based on enhanced firefly algorithm (EFA) using Fuzzy- based grouping and mutation techniques for the detection of network communities. The proposed approach was strengthened with a smart population method. The algorithm also detected network communities where the number of communities was not initially known. The results were compared with other algorithms like MOGA-Net, Bondel, RN,CNM, Infomod and original firefly algorithms. The implementation and testing of EFA on real world and other synthetic data networks showed its efficiency in finding the different communities in large networks.

Shang et al. [29] proposed a community detection method based on modularity and an improved genetic algorithm named as MIGA. The authors also used prior information regarding the number of detecting communities. MIGA used simulated annealing algorithm for local searching. Thus, in short MIGA Algorithm can be summarized in three steps i.e.

- The objective function for community detection is modularity.
- The preceding information of detecting communities i.e. the information about the communities is initially known.
- Simulated annealing algorithm for local searching.

The algorithm was compared with memetic algorithm (ME) and genetic algorithm (GA) for both real-world and computer-generated data networks. The results shown a lesser computational complexity of MIGA in comparison with ME and GA.

Honghao et al. [6] proposed an Ant Colony Optimization (ACO) method for detecting communities in networks using max-min ant system method for community detection. There were two important points of their method:

- Solution in the form of locus-based adjacency in which communities are taken as linked components of networks.
- The structural information was included in the algorithm and thus a new heuristic was proposed, based on association between nodes.

The algorithm was tested on four real-life network and LFR bench mark and results showed the great potential of algorithm in finding communities in networks. This algorithm only deals with static networks.

Hafez et al. [3] used Artificial bee colony (ABC) optimization technique to solve the community detection problem. The objective functions used in this algorithm were divided into two categories:

- The first category of objectives consisted of Conductance, Internal Density, Normalized Cut, Cut Ratio, Expansion, Average-ODF, Flake-ODF and Maximum-ODF.
- The second category of objectives consisted of Modularity, Community Fitness and Community Score.

The solution of the community detection problem was obtained by minimizing the first category and maximizing the second category. The method used is locus-based adjacency for representation of communities in ABC algorithm and thus has advantage that it makes the algorithm to automatically detect the communities. The authors had tested the algorithm for real-world data as well as online social network like facebook and found that the algorithm is efficient in case of accuracy and efficiently detect the communities as well.

Gong et al. [19] proposed fast multi-level memetic algorithm for community detection named MLCD that uses genetic algorithm (GA) with multi-level learning strategies. The proposed hybrid global-local search algorithm used Label propagation method for updation of community identifier of each vertex at each procedure. The results showed a lesser computation time in comparison with original memetic algorithm.

Table 2.2 shows the summarization of various evolutionary approaches:

Authors	Year	Proposed Method	Fitness / Objective Function	Community Detection	
				For Overlapping / Non-Overlapping Networks	In Static / Dynamic Networks
C. pizzuti	2008	Genetic Algorithm in Network's Community Detection (GA-Net)	Community Score and NMI	Non-Overlapping	Static
C. Pizzuti	2009	Multiobjective Based Genetic Algorithm for Network Communities (MOGA-Net)	Modularity and NMI	Non-Overlapping	Static
Amiri et al.	2012	Multiobjective Enhanced Firefly Algorithm with Fuzzy Based Clustering	Multiobjective Optimization Problem (MOP) Function	Non-Overlapping	Static
Shang et al.	2013	Modularity Based Improved genetic Algorithm (MIGA)	Modularity and NMI	Non-Overlapping	Static
Honghao et al.	2013	Ant Colony Optimization (ACO)	Modularity and NMI	Non-Overlapping	Static
Hafez et al.	2014	Artificial Bee Colony Optimization	Conductance, Expansion, Internal Density, Cut Ratio, Normalized Cut, Maximum ODF, Modularity, Community Score and Community Fitness	Non-Overlapping	Static
Ma et al.	2014	Multilevel Learning Based Memetic Algorithm (MLCD)	Modularity	Non-Overlapping	Static

Chapter 3

Problem Definition

Given a graph G (V,E) with a set of n vertices (nodes) $V = \{v_1, v_2, \ldots . v_n\}$ and a set of m edges(links) $E = \{e_1, e_2, \ldots . e_n\}$,the graph reflecting the social structure corresponding to the given community is represented with an adjacency matrix A of size V X V such that for any pair of vertices i and j

$$A_{i,j} = \begin{cases} 1 \text{ if an edge between i and j} \\ 0 \quad \text{Otherwise} \end{cases}$$

Where the adjacency matrix, A is symmetric (Assuming the graph as an undirected graph).

The problem of community detection relies on finding the subgraphs i.e. partitioning the graph *G* into n subgraphs $G_1, G_2, \ldots\ldots..G_n$ and $V = G_1 \cup \ldots\ldots \cup G_n$ such that all G_i for all $i \in n$ correspond to the communities of densely linked nodes, with the nodes belonging to different communities being only sparingly connected based on the criteria of optimizing modularity value. The Modularity evaluates the quality of cluster which signify the extent to which a given community partition is distinguished by high number of intra-community connectivity in comparison to inter-community ones [4, 6].

The Modularity (Q) value [2, 25, 29] can be mathematically stated as

$$Q = 1/2m \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(i,j)$$

Where, $A_{i,j}$ is the adjacency matrix, m is the number of edges in network, d_i, d_j are the degree (or strength) of nodes i, j and $\delta(i, j)$ is the function which return 1 when both i, j are in same community, 0 otherwise.

The modularity of a community is a scalar value between -1 and 1 that computes the degree of cohesiveness within community as compared to links between communities [25]. More the modularity value better is the quality of the communities detected.

Chapter 4

Proposed Algorithm for Problem Solution

The work presents a novel nature-inspired algorithmic approach based on Ant Lion Optimizer (ALOCD) for solving the community detection problem .The underlying algorithm proposed by Seyedali Mirjalili [32] utilizes the unique hunting behavior of antlion as shown in Fig 4.1. The algorithm is inspired by of hunting behavior of prey such as the random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps .These features allow the antlions to take the positions of ants and making the antlions move towards better fit ants to find a global best possible solution. The main components of the proposed algorithm are as follows:

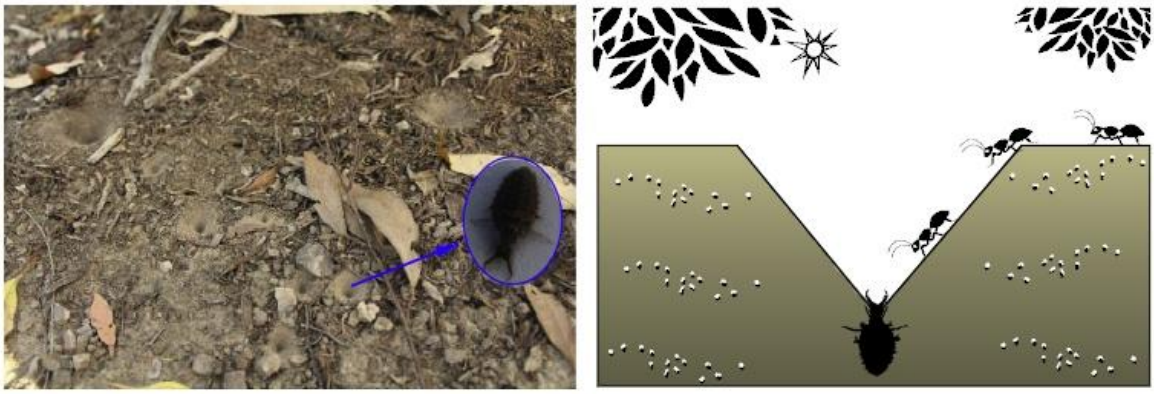

Fig4.1: showing cone shaped pit dig by antlion and their hunting behavior [32]

4.1. Solution Representation

In the proposed approach each solution/antlion in the population is encoded as a collection of n positions $S = [a_1, a_2, \ldots a_n]$ such that each value at location i contains a positive integer, $a_i \in [1, n]$ that interprets the community to which the i-th node belongs. The node i and node j belong to same cluster if the value a_i equals a_j for any set of i and j nodes. One antlion stands for one solution that divides given community structure into sub-community partitions. Fig 4.2 illustrates a solution representation of a social network with 10 nodes using an array data structure

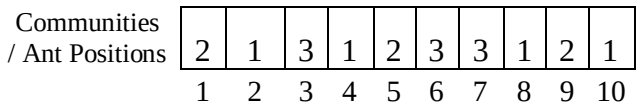

Fig 4.2: Solution Representation

As shown in Fig 4.2, a set of nodes { 2,4,8,10} belonging to given community structure in hand, belongs to the sub-community/cluster labeled number 1 and set {1,5,9} and set {3,6,7} to 2^{nd} and 3^{rd} sub-community number respectively.

4.2. Random Walk of Ants

In every iteration, the position of each ant is updated with respect to a selected antlion on the basis of elite (best antlion obtained so far) and roulette wheel selection operator. The updation of positions is done with the help of two random walks i.e. random walk on the basis of roulette selected antlion and elite antlion.

For a given community of size say 'n' , pick value at random position [1:n] from candidate solution(elite/roulette selection) which represents sub-community number to which that indexed vertex belongs to. Fig 4.3 shows the representation of Candidate Solution before random walk.

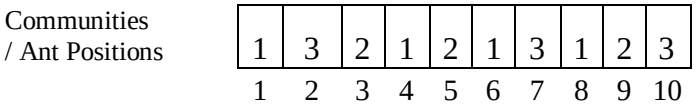

Fig 4.3: Candidate Solution (before random walk)

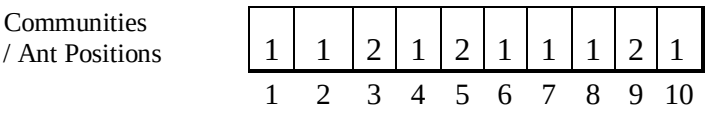

Fig 4.4: Solution after Merging 1^{st} and 3^{rd} Communities

Fig 4.4 shows one step in random walk procedure by taking a random position say 4^{th} in candidate solution and generates new solution after merging 1^{st} and 3^{rd} communities. If the newly generated solution gives better modularity than candidate solution, the candidate solution is updated. This step of random walk is applied recursively depending upon the size of the trap. For assuring exploitation of search space, the radius of updating ant's positions is

contracted. This step is modeled by decreasing the random walk rate as the *iteration* value approaches *Num_of_gen* value.

4.3. Updating the position of Ants

During each iteration of the algorithm, the movements of all the ants are affected by elite as well as selected antlion as every ant randomly walks around a selected antlion by the roulette wheel and the elite concurrently .This concurrent effect of both selected and elite antlion on the movement of ant is modeled using *Update_Ant_pos()* procedure as shown in Fig 4.5.

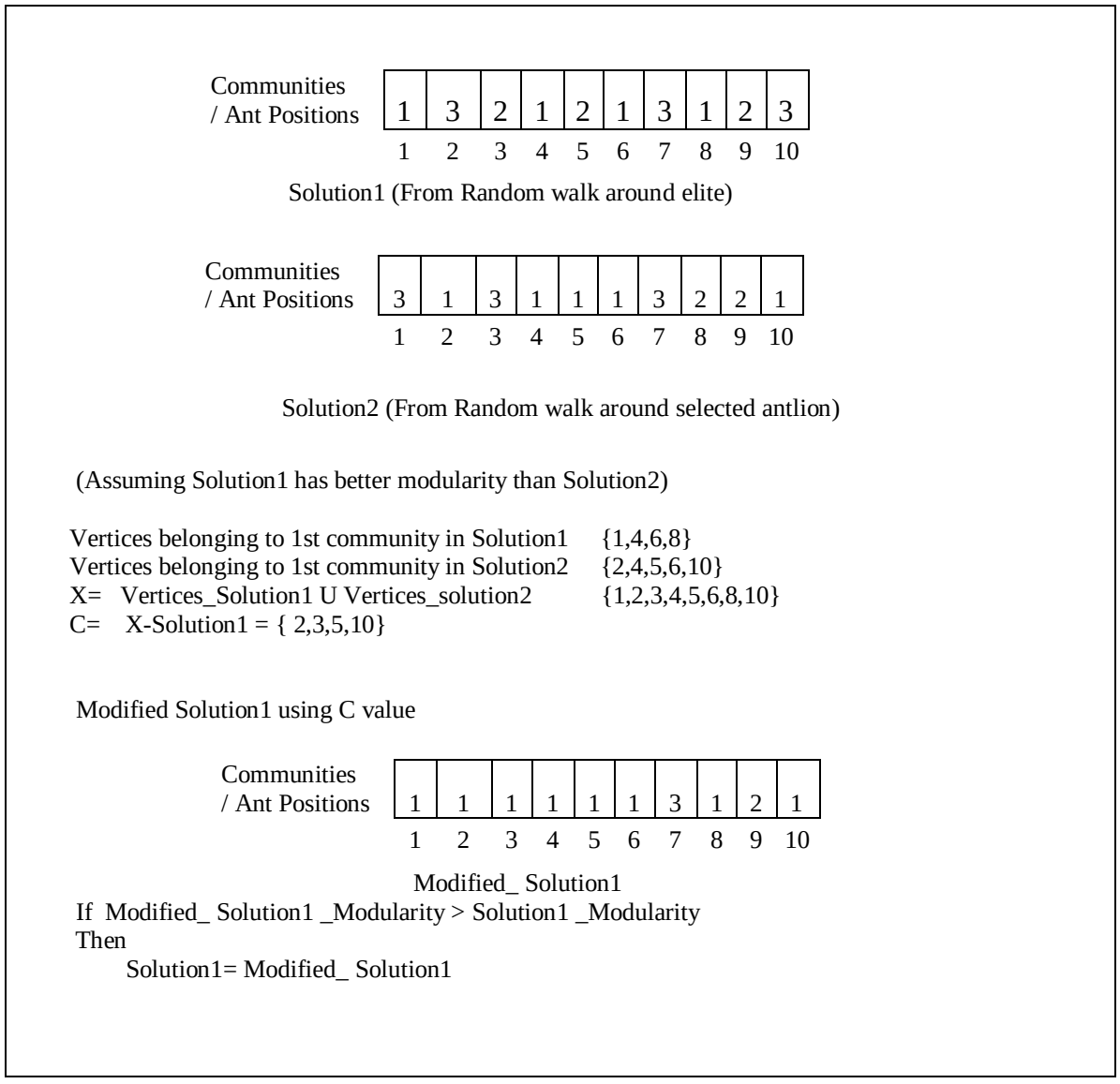

Fig 4.5: One Step of *Update_Ant_pos()* procedure

The steps shown in Fig 4.5 are repeated for all the possible communities while retaining better solutions in every repetition. The algorithm for the ALOCD approach has been illustrated by the following pseudo code:

Pseudocode of proposed ALOCD algorithm

Community_AntLionDetector (Input_File, Num_Nodes, Num_nets, Numclass , class, Pop_size, Num_of_gen)

Input : *Read the Benchmark files of communities*

Variables: *pop_ant, pop_antlion array of structures of solution of size (Pop_size X Num_Nodes), Merge_solution array of structures of solution of size ((2 X Pop_size) X Num_Nodes)*

Class- true community of community structure, Numclass -Total no.of true communities

Output: *Set of Best Communiites*

Begin

```
[adj_array]=Create Netlist(File_IO, Netlist, A, Num_Nodes, Num_nets)
[pop_ant]=Initalise(Pop_size,Num_Nodes,adj_array)
[pop_antlion]=Initalise(Pop_size ,Num_Nodes,,adj_array)
Set iteration:=1
 While (iteration <= Num_of_gen)
            [Elite]=Findbest(pop_antlion,Pop_size,Num_Nodes)
          For every ant  i=1:Pop_size
                        For j=1:Pop_size
                        w(j)=pop_antlion(j).fit
                 Endfor
                choice = RouletteWheelSelection(1./w,Pop_size)
       If choice==-1
                        choice=1
       endif
```

```
                    roulete_antlion=pop_antlion(choice)
                    RA = Random_walk (Elite,Num_Nodes,adj_array)
                    RB = Random_walk (roulete_antlion,Num_Nodes,adj_array)
                Cross_R= Update_Ant_pos(adj_array,RA,RB,Num_Nodes)
                pop_ant(i)=Cross_R
                If pop_ant(i).fit>Elite.fit
                        Elite= pop_ant(i)
                Endif
        Endfor
        Merge_solution =Merge_Sort(pop_ant,pop_antlion, Pop_size,Num_Nodes)
        [pop_antlion ]=Update_solution(Merge_solution, Pop_size,Num_Nodes)
  Endwhile
[C]= pop_antlion(1).bit
[NMI] = Compute_NMI(class, C)
Print: 'nmi ', NMI
Print: 'Optimized Community ', pop_antlion(1).bit

End
```

A solution is encoded as a random permutation of n positions of ant/antlion representing the communities of given input net list file pertaining to the social network. The initial solutions are preprocessed by randomly picking a vertex V_i and finding the immediate adjacent to that vertex sayV_j. The community value of V_i is allocated to that of V_j. This process is repeated for rest of vertices until sufficient number of solutions for initial population is generated. After initialization, the best antlion (elite) is chosen from the generated population on the basis of modularity value of antlion. More modularity contributes to better antlion. The quality of solutions/antlions is improved through random walk. After random walk, if the new ants have high fitness than antlions, then antlion new positions will become positions of the ants (for imitating the process of catching the prey and the antlion is then required to change its position to the most recent position of the hunted ant to boost its chance of catching new prey. In each iteration, the antlion with highest fitness is substituted as best antlion (elite). The process is repeated until the optimal solution is

obtained. At final stage, the first solution of the population with best modularity value is printed along with their NMI value.

Chapter 5

Experimental Results

5.1. Real-world Networks used as Benchmark

The proposed ALOCD approach is implemented using matlab 7.11.0 (R2012a) on intel core i5 processor, with 4 GB RAM under 64-bit Operating System. The work is tested on Zachary's Karate Club, Bottlenose Dolphins, Books about US politics and American college football network [20] benchmarks and results are compared with the ACO and EFF [6, 4].

Zachary's karate club as depicted in Fig 5.1 is mostly used as a community detection networks. This club consists of 34 members which represents as 34 nodes and the relationships between the club members is represented as edges.

Dolphins network as depicted in Fig 5.2 is proposed by Lusseau after observing the behavour of dolphins. The 62 dolphins in this network is represented as nodes and the connections between these dolphins are represented as edges.

Girvan and Newman proposed a network depicted in Fig 5.3 that represents different football teams as nodes and the matches between any two teams as edges which was named as American college football club.

The books about US politics is a network shown in Fig 5.4 published in 2004. Newman divided this network in three communities on the basis of reviews and descriptions.

Table 5.1 shows the basic features of the real world networks and their true number of community structures [20]

Benchmark Networks	Nodes	Edges	True Communities
Zachary's Karate Club	34	78	2,4
Bottlenose Dolphins	62	159	2,4
Books about US politics	105	441	3
American College Football	115	613	12

5.2. Normalized Mutual Information

The performance of ALOCD approach is evaluated using *Normalized mutual information* (NMI) [4,7,8,11] that quantifies the similarity between the detected and true community structure .

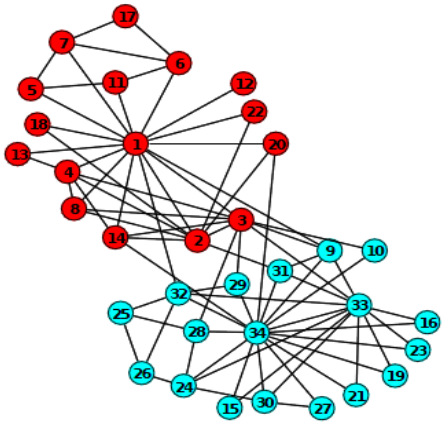

Fig 5.1: Zachary Karate club [20]

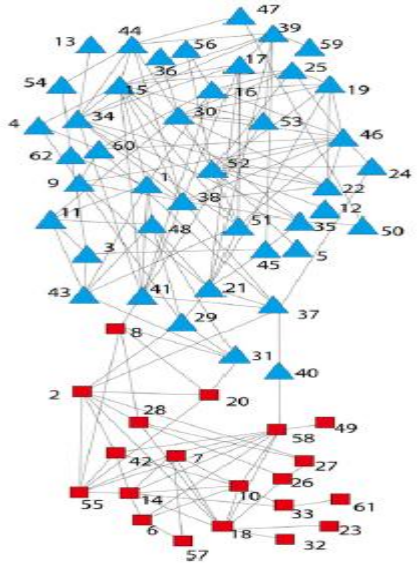

Fig 5.2: Bottle Dolphins [29]

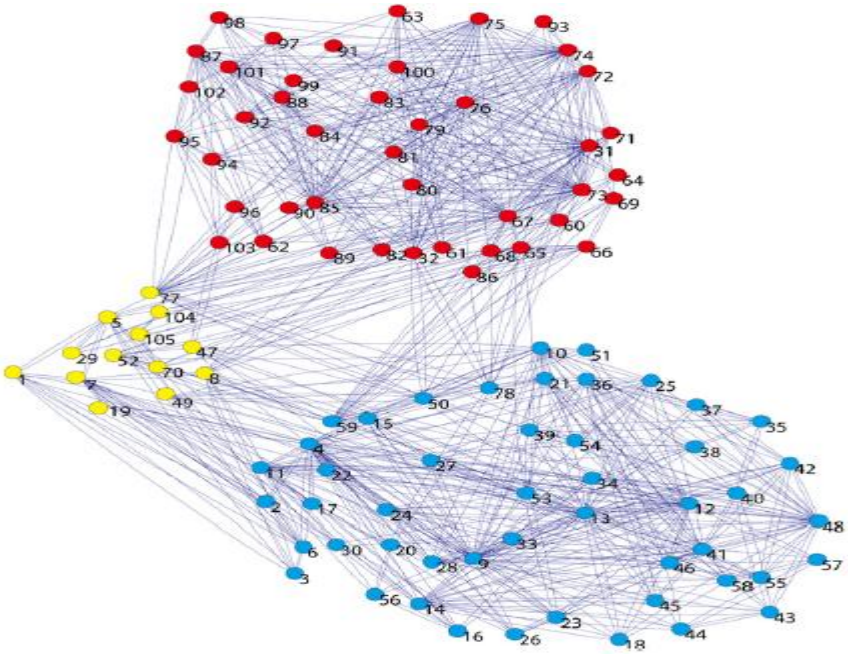

Fig 5.3: Books about US politics [29]

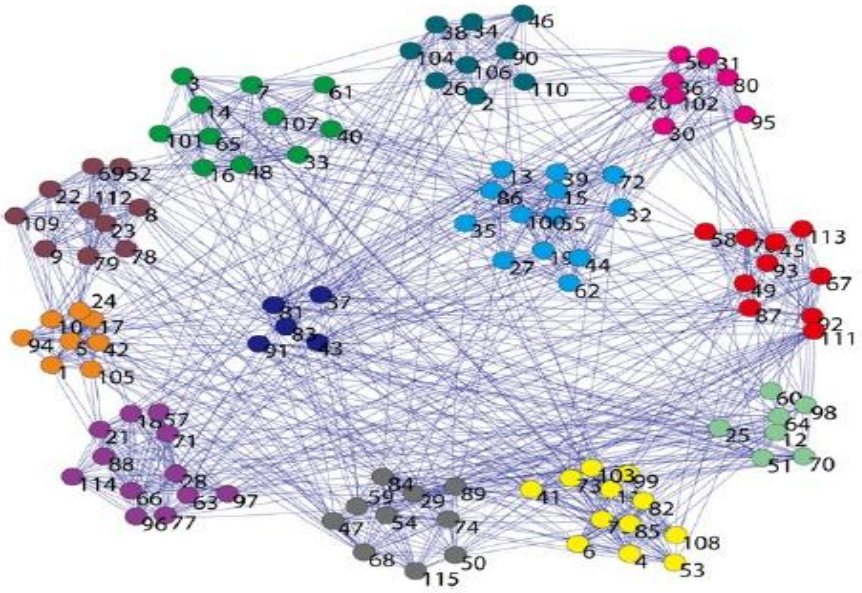

Fig 5.4. American Football club [29]

NMI denoted as I (X, Y) is calculated using the following formula as shown below:

$$I(X, Y) = -\frac{2\sum_{i=1}^{C_X}\sum_{j=1}^{C_Y} C_{ij} \log\left(\frac{C_{ij}N}{C_i . C_j}\right)}{\sum_{i=1}^{C_X} C_i . \log(C_i/N) + \sum_{j=1}^{C_Y} C_{.j} \log(C_{.j}/N)}$$

Where X and Y denotes two network structures, C - the confusion matrix. C_{ij} - the number of nodes present in community i of X as well as in community j of Y , $C_X(C_Y)$ - the number of classes in part X and Y, $C_i . C_j$ - the number of elements in row i (column j) of C, N -the total number of nodes in networks.

The greater the value of NMI, the more similar the true and detected communities are i.e. better the solution quality. For both communities being same, the NMI value equals 1 and NMI=0 signifies different communities.

Table 5.2 shows the results of implementation of the proposed approach over 10 different runs for a given set of benchmark networks. The algorithm computes the NMI value and determines the total number of communities in each run, as shown in table 5.2.

Table5.2 represents the NMI values and number of communities for multiple runs of ALOCD approach

Benchmark Networks		Number of Runs									
		1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**8th**	**9th**	**10th**
Zachary's Karate Club	**NMI**	0.732	0.732	0.592	0.494	**1.000**	0.732	0.516	1.000	0.732	0.732
	Number of Communities	2	2	4	4	**2**	2	8	2	2	2
Bottlenose Dolphins	**NMI**	0.502	0.534	0.699	0.798	0.698	0.835	**0.887**	0.802	0.513	0.756
	Number of Communities	5	3	2	3	4	2	**2**	3	6	4
Books about US politics	**NMI**	0.696	0.508	0.449	0.506	0.726	**0.733**	0.431	0.668	0.705	0.534
	Number of Communities	4	3	10	15	4	**3**	7	5	3	9
American College Football	**NMI**	0.537	0.608	0.503	0.804	0.775	0.795	0.665	**0.850**	0.611	0.759
	Number of Communities	17	16	14	13	12	13	22	**12**	20	14

***Bold faced values represent best results of ALOCD approach for respective benchmark networks**

From the tabulated values as shown in Table 5.2, it is concluded that the proposed approach is competent to detect 100% community structure information for Zachary's karate network [20]. Fig 5.5 reveals NMI value equal to 1 at 5th and 8th run of the program execution with the number of communities same as that of true community structure for Zachary's karate network.

Fig 5.6, 5.7 and 5.8 shows the NMI values of ten runs of ALOCD on Bottlenose Dolphins, Books about US politics and American Football Club benchmark networks [20] respectively. The 7th run of the implementation of ALOCD approach, the no of communities found for bottlenose dolphins is equal to true number of community structures with NMI value of 0.887 as shown in Fig 5.6.

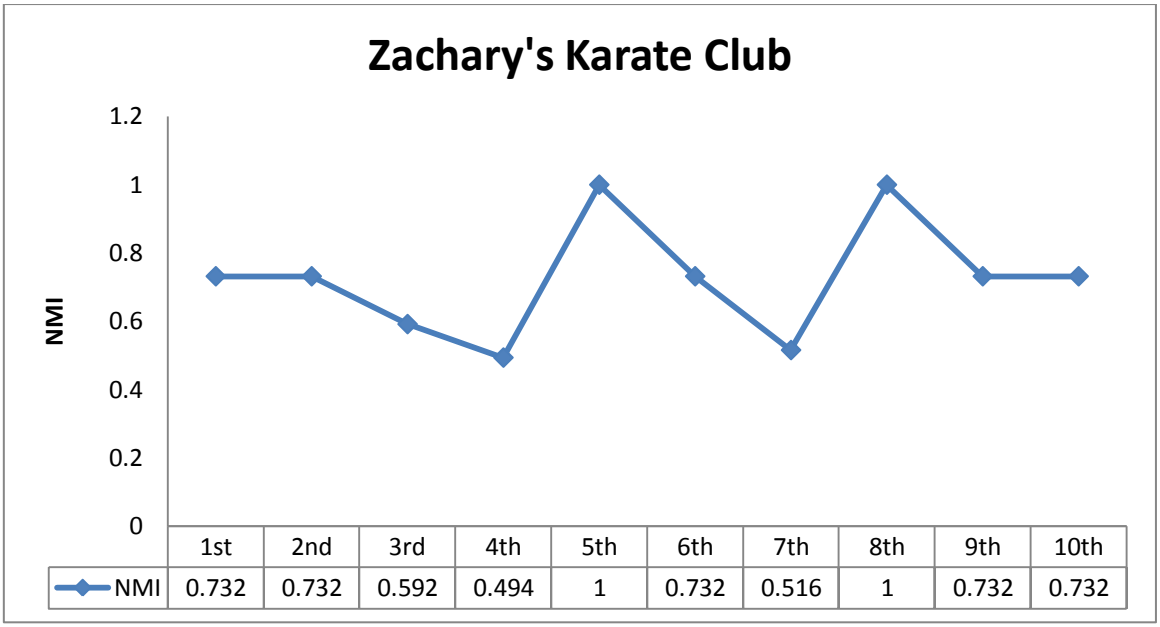

Fig 5.5: NMI values of ten runs of ALOCD on Zachary's Karate Club

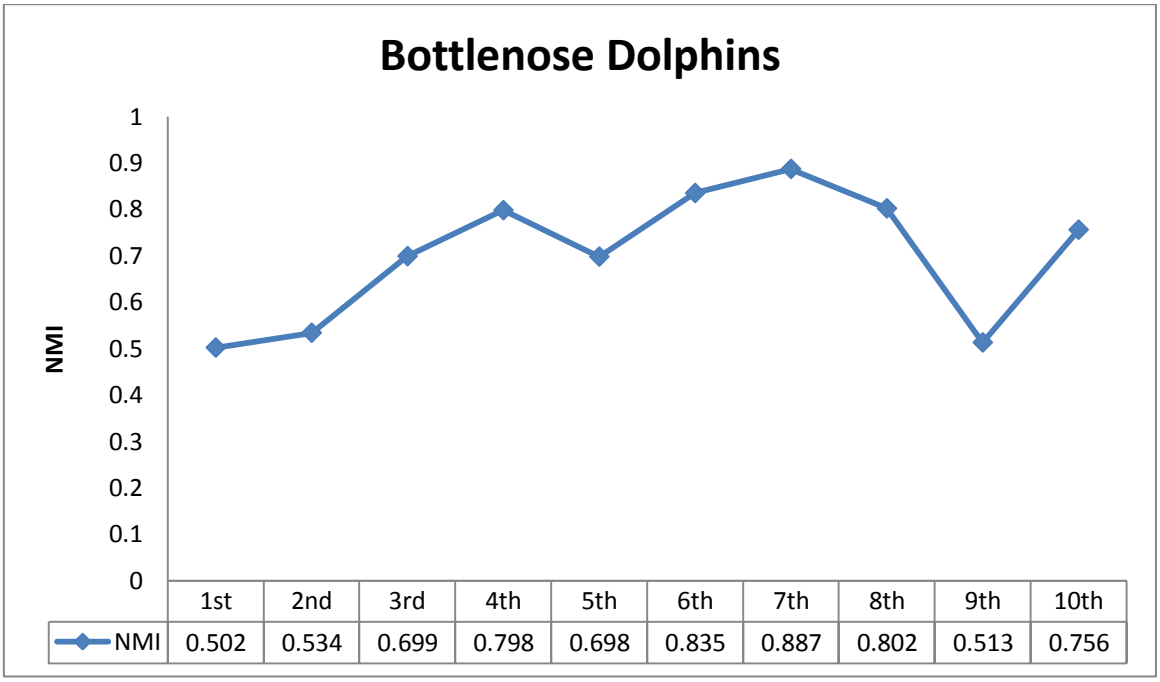

Fig 5.6: NMI values of ten runs of ALOCD on Bottlenose Dolphins

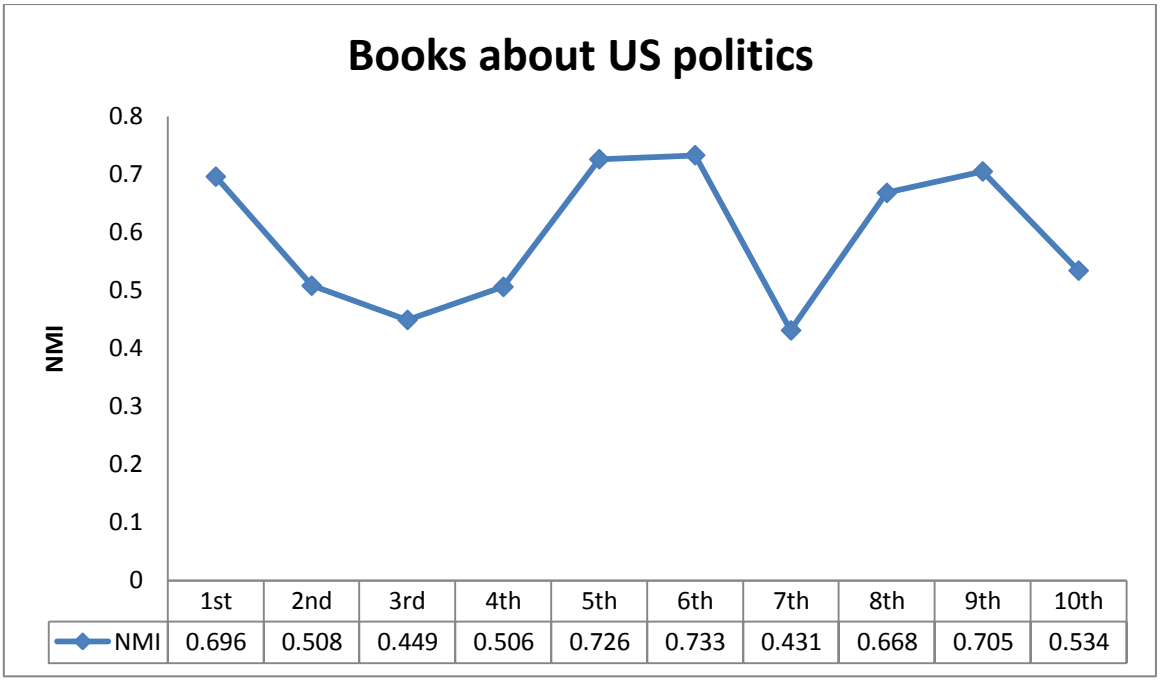

Fig 5.7: NMI values of ten runs of ALOCD on Books about US politics

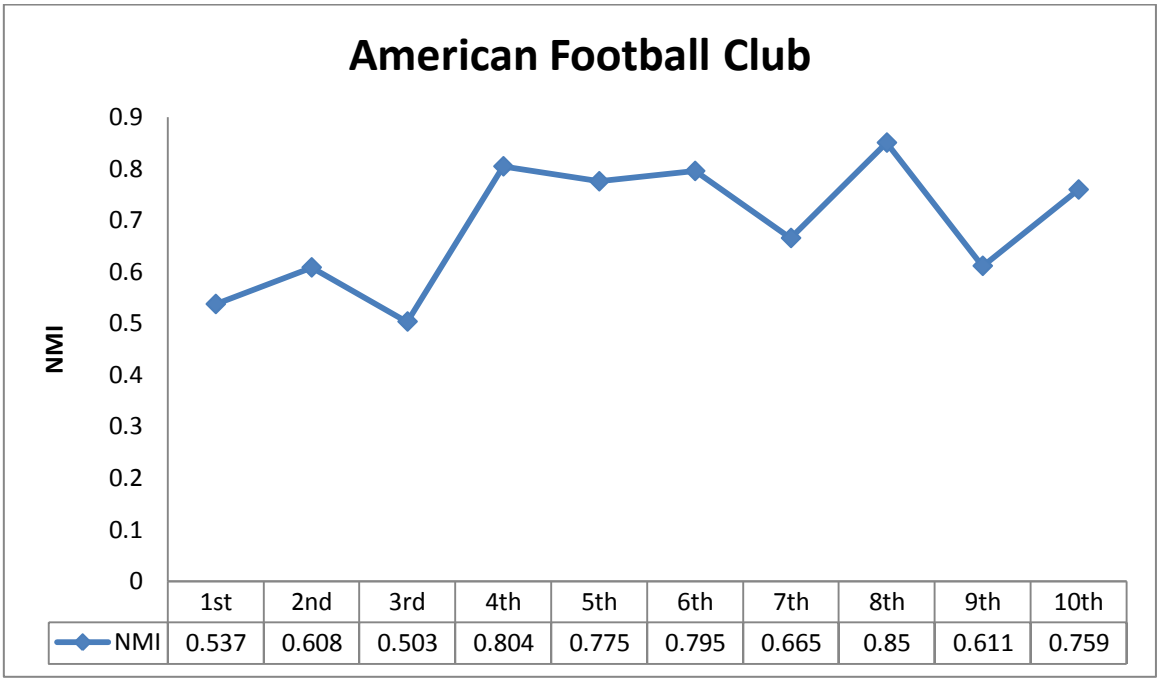

Fig 5.8: NMI values of ten runs of ALOCD on American Football Club

From Fig 5.7 it is clear that the number of communities found for Books about US politics is equal to the number of true communities with NMI value of 0.733 at 6^{th} run of program execution. Fig 5.8 depicts same number of communities as that of actual one at 8^{th} run of implementation for American Football with NMI value of 0.850.

Table 5.3 shows the count and nodes of wrong communities along with best NMI of all networks

Benchmark Networks	NMI	Wrong communities	Nodes in Wrong communities
Zachary's Karate Club	1.000	-	-
Bottlenose Dolphins	0.887	1	[20]
Books about US politics	0.733	6	[8, 15, 28, 46, 54, 89]
American College Football	0.850	13	[10, 18, 23, 32, 39, 47, 52, 54,65, 78, 82, 86, 101]

The table 5.3 clearly shows the very good performance of ALOCD approach for Zachary's Karate Club Benchmark Networks. The solutions found by ALOCD approach split the communities into 2,3,12 clusters with 1,6,13 nodes misplaced for Bottlenose Dolphins, Books about US politics and American College Football respectively.

Table 5.4 shows the comparison of best NMI values of ALOCD, ACO [6] and EFF [4] algorithm and also shows at best NMI the respective communities found in all networks

Benchmark Networks	NMI and No. of Communities	ALOCD	ACO	EFF
Zachary's Karate Club	NMI	1.000	0.687	0.998
	Communities	2	2	4
Bottlenose Dolphins	NMI	0.887	0.587	0.988
	Communities	2	2	4
Books about US politics	NMI	0.733	0.560	0.599
	Communities	3	2	4
American College Football	NMI	0.850	0.890	0.798
	Communities	12	12	11

The results of the proposed approach are compared with ant colony optimization (ACO) [6] and enhanced firefly algorithm (EFF) [4] in terms of best NMI values obtained. From Table 5.4 and Fig 5.9, it is clear that for Zachre's karate network, the proposed ALOCD approach detects 100% true community structure, for Dolphins the algorithm has obtained best normalized mutual information of 0.887 close to EFF best NMI value, while the NMI of ACO was 0.798 and for Books about US politics the algorithm has highest NMI value i.e. 0.733 respectively in comparison with other approaches.

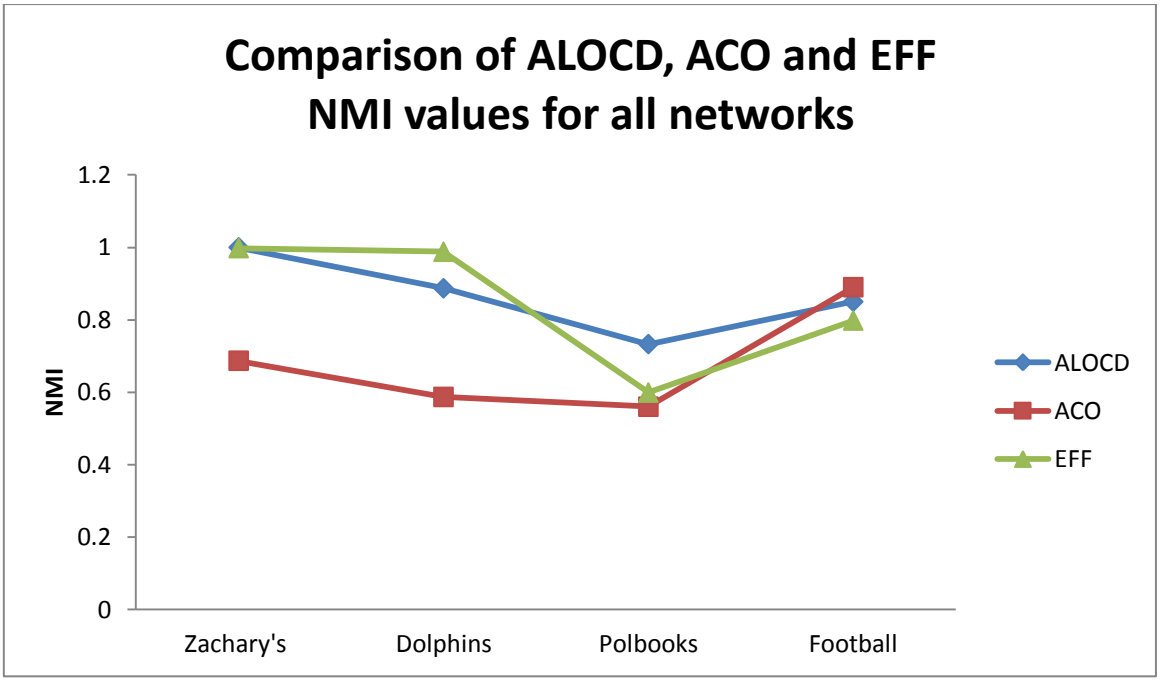

Fig 5.9: Comparison of best NMI values of ALOCD with ACO and EFF algorithm NMI values for four real world networks

On the American College Football network, ALOCD obtained best normalized mutual information of 0.850 close to ACO best NMI value, while the NMI of EFF was 0.798 as shown in table 5.4. The proposed approach is able to detect the true number of communities with NMI value close enough to NMI value of true community structure.

Chapter 6

Conclusion and Future Scope

In this thesis, various community detection algorithms and also the various evolutionary approaches have been described that was applied for solving the community detection in networks. The proposed approach optimizes the modularity function and is able to identify densely connected groups of nodes having sparse interconnections. The performance of the algorithm is measured in terms of Normalized Mutual Information (NMI) function.

The algorithm is tested on four real-world networks i.e. Zachary's Karate Club, Bottlenose Dolphins, Books about US politics and American Football Club. The experimental results show 100% community structure information detection for Zachary's Karate club. The approach also gives promising results for other networks by finding the true number of communities with NMI values close to true community structures.

The ALOCD approach is also compared with Ant Colony Optimization (ACO) and Enhanced Firefly algorithm (EFF) over given set of benchmarks. The approach outperforms EFF and ACO for Zachary and Books about US politics and produces results better than ACO and comparable to EFF for Dolphins and also produces results better than EFF and comparable to ACO for American Football Club.Thus, ALOCD algorithm produces efficient results for these four real world networks described above.

In future, the work can be further extended for other bigger real world networks like Facebook, Twitter etc. This nature inspired approach can be improved to solve the problem of community detection for overlapping communities and dynamic community detection.

References

[1] A. Kouznetsov and M. Tsvetovat, "Social Network Analysis for Startups," O'Reilly Media, September, pp. 1-192, 2011.

[2] A.Clauset, M. E. J. Newman and C. Moore, "Finding community structure in very large networks," *Physical Review E*, 70, 0066111, 2004.

[3] A. I. Hafez, H. M. Zawbaa, A. E. Hassanien, A. A. Fahmy, "Networks Community Detection Using Artificial Bee Colony Swarm Optimization," *Advances in Intelligent Systems and Computing*, vol.303, pp.229-239, 2014.

[4] B. Amiri, L. Hossain, J. W. Crawford and R. T. Wigand, "Community Detection in Complex Networks: Multi–objective Enhanced Firefly Algorithm," *Science Direct, Knowledge Based Systems, Elsevier*, vol.46, 2013.

[5] C. Bichot and P. siarry, "Graph Partitioning," *Iste Series, Wiley*, pp. 1-320, 2011.

[6] C. Honghao, F. Zuren and R. Zhigang, "Community detection using Ant Colony Optimization," *IEEE Congress on Evolutionary Computation (CEC),* pp.3072-3078, 2013.

[7] C. Pizzuti, "GA-Net: A Genetic Algorithm for Community Detection in Social Networks," *Lecture notes in Computer Science, Springer,* PPSN X, vol.5199, pp. 1081–1090, 2008.

[8] C. Pizzuti, "A Multiobjective Genetic Algorithm to Find Communities in Complex Networks," *IEEE Transactions on Evolutionary Computation,* vol.16, no.3, 2012.

[9] D. Emanuel and A. Fiat," Correlation clustering - minimizing disagreements on arbitrary weighted graphs," *Lecture Notes in Computer Science, Springer Berlin Heidelberg,* pp. 208-220, 2003.

[10] E. Ferrara and G. Fiumara, "Topological Features of Online Social Networks," *Communications on Applied and Industrial Mathematics*, vol. 2, no. 2, pp. 1-20, 2011.

[11] E. L. Martelot and C. Hankin, "Fast Multi-Scale Detection of Relevant Communities in Large Scale Networks," *The Computer Journal Oxford University Press*, 2013.

[12] E. L. Martelot and C.Hankin, "Fast Multi-Scale Detection of Overlapping Communities using Local Criteria," *Computing, Springer*, 2014.

[13] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto and D. Parisi,"Defining and identifying communities in networks," *in Proceedings of the National Academy of Sciences of the United States of America*, pp:2658–2663, 2004.

[14] G. Tibély and J. Kertész," On the equivalence of the label propagation method of community detection and a potts model approach," *Physica A: Statistical Mechanics and its Applications*, 2008.

[15] J. Bruhn," The Sociology of Community Connections," *in Springer Science + Business Media B.V*, pp.1-328, 2011.

[16] J. Liu, "Comparative Analysis for k-Means Algorithms in Network Community Detection," *Advances in Computation and Intelligence, Springer,*vol. 6382, pp 158-169, 2010.

[17] J. Kleinberg,"An impossibility theorem for clustering," *Advances in neural information processing systems*, pp. 463-470, 2003.

[18] J. Sima, S. E. Schaeffer," On the np-completeness of some graph cluster measures," *in Proceedings of the 32nd conference on Current Trends in Theory and Practice of Computer Science, Springer-Verlag* , pp.530-537, 2006.

[19] L. Ma, M. Gong, J. Liu, Q. Cai and L. jiao, "Multi-level learning based memetic algorithm for community detection," *Science Direct, Applied Soft Computing, Elsevier,* vol.19, pp.121-133, 2014.

[20] M. E. J. Newman (Last modified: April 19, 2013). [Online]. Available: http://www-personal.umich.edu/~mejn/netdata/ [Accessed on: May 28, 2015]

[21] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, 69:026113, Vol. 69, 2004.

[22] M. E. J. Newman," Fast algorithm for detecting community structure in networks," *Physical Review E*, 69:066133, 2004.

[23] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, 74, 036104, 2006.

[24] M. J. Rattigan, M. Maier and D. Jensen," Graph clustering with network structure Indices" *In Proceedings of the 24th international conference on Machine learning,* pp. 783-790, 2007.

[25] M. E. J. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no.23, June 2006.

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

[27] P. Schetz, A. Calflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, 77, 046112, 2008.

[28] R. Missaoui and I. Sarr, "Social Network Analysis - Community Detection and Evolution", *Lecture Notes in Social Networks, Springer*, pp. 1-272, 2015.

[29] R. Shang, J. Bai, L. Jiao and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Science Direct, Physica A: Stastical Mechanics and its Applications, Elsevier,* vol.392, pp.1215-1231, 2013.

[30] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486 pp. 75-174, 2010.

[31] S.Kirkpatrick, M. P. Vecchi and C. D. Gelatt, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[32] S. Mirjalili, "The Ant Lion Optimizer," *Science Direct, Advances in Engineering Software, Elsevier,* vol.83, pp.80-98, 2015.

[33] U. Luxburg," A tutorial on spectral clustering." *Statistics and Computing, Springer*, vol. 17, pp. 395-416, 2007.

[34] U. N. Raghavan, R. Albert and S. Kumara," Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, 76:036106, 2007.

[35] V. D. Blondel, J.L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment,* vol.1742, no.10, 5468, P10008, 2008.

[36] Y. Hu, H. chen, P. Zhang, M. Li, Z. Di and Y. Fan," Comparative definition of community and corresponding identifying algorithm," *Physical Review E*, 78:026121, 2013.

List of Publications and Video Link

[1] A. Mahajan, M. Kaur," Community Detection in Complex Networks: A novel nature-inspired algorithmic approach based on Ant Lion Optimizer (ALOCD) for solving the community detection problem," 2015 (Communicated).

[2] A. Mahajan, M. Kaur," Various Approaches for Community Detection in Complex Networks: A Glance," 2015 (Communicated).

[3] https://www.youtube.com/watch?v=wleMoZCHsFo (Video Link)