Community Detection Using Nature Inspired Algorithm

Sunita Chand and Shikha Mehta

Abstract Community detection in social networks has become a dominating topic of the current data research as it has a direct implication in many different areas of importance, whether social network, citation network, traffic network, metabolic network, protein-protein network or web graph etc. Mining meaningful communities in a real-world network is a hard problem owing to the dynamic nature of these networks. The existing algorithms for community detection depend chiefly on the network topologies and are not effective for such sparse graphs. Thus, there is a great need to optimize those algorithms. Evolutionary algorithms have emerged as a promising solution for optimized approximation of hard problems. We propose to optimize the community detection method based on modularity and normalized mutual information (NMI) using the latest grey wolf optimization algorithm. The results demonstrate the effectiveness of the algorithms, when compared with other contemporary evolutionary algorithms.

Keywords Community detection • Grey wolf optimization • Modularity • NMI

1 Introduction

Humans have always preferred to live in society by making groups to form small or big communities. The communities could evolve based on race, colour, caste, religion, gender, job category, interest, hobbies etc. Nowadays, with the rapid advancement in the speed of the Internet and its reduced cost, we try to socialize through social networks, despite being separated by long distances across the globe. This has been made possible because of the social networks available to us. These networks have become the backbone of virtual socialization and help people to interact and share their latest updates with their friends and relatives. Thus, social

S. Chand (✉)
Hansraj College, University of Delhi, New Delhi, India
e-mail: sunitamk@gmail.com

S. Mehta
JIIT Noida, Jaypee University, Noida, India
e-mail: shikha.mehta@jiit.ac.in

H. Banati et al. (eds.), *Hybrid Intelligence for Social Networks*,
DOI 10.1007/978-3-319-65139-2_3

networks such as Facebook, Twitter, MySpace and LinkedIn have occupied a major share of internet usage. These social networks have not only helped us to be friends with others, but have become an important tool for propagating ideas and our views on a large scale. These social networks have emerged as an important tool for politicians for their mass campaigning during the election period. It has also helped retailers and businesses to promote their products. At the same time, ordinary people have opted for social media to raise their voices against any social crime or injustice. Considering the importance of social networks, there is a great need to analyze these networks from various aspects. Social network analysis may be done at different levels, i.e. at a node level, at a group level or at the network level [7]. We may need to identify the group or cluster to which a node belongs. Sometimes we may also need to analyze the social network to find the most prominent personality on the network. The one with the maximum number of followers across all the communities may give us an indication of how important a person is in the group. Analogous to our social physical groups or communities, the virtual network of nodes on social media may also be categorized into communities. Recognizing these communities could be of particular interest to researchers and analysts. For instance, finding the community of a collaboration network may help us to discover groups of researchers working in the similar domain or having expertise in a similar field. In social networks, community detection helps by recommending a person to be a friend of a person who may belong to a certain group, e.g. a group of ex-school friends, a group of office colleagues, a family group, or a group of close friends. We may also recommend a person to join a particular group based on the links that exist between that person and a member of that group. Hence, investigating community has become a hot topic of research these days. The analysis may be based on several parameters, e.g. degree centrality, betweenness, closeness, information and rank. Community detection (CD) in a social network has come forth as a fundamental tool for social network analysis that lays the foundation for several higher-level analytics. It is associated with partitioning a social network into various cohesive subgroups or clusters in such a way that there is a strong cohesion or interconnection among the nodes of a subgroup and weak cohesion among the nodes of different subgroups.

The process of detecting communities within large networks, if automated, could prove to be very helpful for the network analysts. For instance, communities in a web graph may represent a group of web sites concerned with related topics, whereas communities in an electronic circuit or a biochemical network may correspond to some sort of functional unit in that network [9]. Social networks may be defined as vast networks of vertices representing individuals or entities, and the edges connecting the vertices as the interaction between them. They can be represented by a graph $G(Vn, Em)$ where V is the set consisting of n nodes and E is the set of m edges connecting them. Practically, in social networks, "n" and "m" keep changing dynamically, leading to increased complexity. The value of "n" may be in hundreds, thousands or even millions, as in Facebook, Twitter etc. Graph problems such as clique, the independent set problem, vertex colouring, edge colouring, travelling salesman problem, graph isomorphism and graph partitioning are considered to be NP hard problems. As social networks can also be represented as graphs, analysis of social networks is also considered to be a nondeterministic polynomial time (NP) hard problem [9]. With the rapid industrialization and technological transformation, there is a great need for optimization in every aspect of problem solving. As the number of dimensions, size of problems and the number of variables increase, the problems become complex and difficult to optimize. Social networks analysis is one such problem where there is a great need for optimization. In this chapter, we attempt to present the computer algorithms for the optimized extraction of communities from social network data. We present an introduction to social networks and the techniques for social network analysis in Sect. 2 of this chapter. Section 3 concentrates on nature-inspired algorithms and their applications. Section 4 gives a detailed insight into community detection in social networks and various solutions to community detection. We present a novel approach to community detection in Sect. 5 of this chapter. Sect. 6 concludes this chapter.

2 Social Networks and Their Analysis

The network has been defined as a set of entities or nodes that are linked by edges connecting them. These nodes may be used to represent anyone or anything. An edge connecting any two nodes represents the relation or the link existing between those nodes [2]. These edges can have characteristic properties such as direction, which signifies the flow of information, and the weight, signifying the cost of the link. The cost may be the distance or the financial cost. Examples of networks include cellular phone networks [33], biological networks, protein-protein interaction networks, communication and transportation networks, collaboration networks, the Internet, the world-wide-web and social networks [15].

2.1 Social Networks

Social networking services, also known as social networks or social media, provide an online platform that facilitates people building social relationships or social networks of friends.

The nodes in social networks represent the people and the edges signify the type of social interaction existing between them. For instance, the relationship may be the common personal interest, friendship, religion, activities, backgrounds, career, real-life connections etc. These sites are aimed at connecting individuals and creating clusters. The social networks may be categorized into three types:

1. **Socializing social networks:** These networks are primarily used for socializing with friends and family members (for example, Facebook and MySpace).
2. **Networking social networks:** These are used primarily for non-social interpersonal interaction and have a specific target audience (e.g. LinkedIn (Business)).

3. **Social navigation network services:** These sites are aimed at helping users to find specific resources and information (e.g. Jeevansathi.com for match-making, Goodreads for books, Last.M for music, Match.com for dating, Yahoo personals, HASTAC, Ning, TermWiki for education and PatientsLikeMe for healthcare).

With the rapid increase in social networking, other applications of social media are also emerging, such as:

1. **Real-time web:** Based on the pattern of live radio and television broadcast, these social media wherein the users can broadcast what they are doing (Twitter, Clixtr and Facebook with "Live feed")
2. **Location based web:** Merges business solutions and technologies, e.g. cloud computing with social networking concept (e.g. Foursqures, Gowalla, Yelp etc).

In recent years, traffic from social networks has occupied a huge share of the overall traffic on the Internet.

2.2 Representation of Networks

Like other networks, a social network may be mathematically modelled as a graph $G = (V, E)$, where the set V represents the set of individuals and the set E represents the set of edges. The total number of individuals can be represented by $n = |V|$ and the total number of edges in the network by $m = |E|$. A community in a network is a subgroup of the vertices that have a high density of edge interaction within them, and a relatively lower density of edge interaction between them. This problem of dividing the network into k unknown communities in a network, may be formulated as partitioning the graph into k partitions that have dense intra-connection but sparse inter-connections. As a network is represented as a graph, we represent the network consisting of N nodes using a $N * N$ adjacency matrix A, where

$$A(i,j) = \begin{cases} 1 & \text{if node } i \text{ is connected to node } j, \\ x & \text{if node } i \text{ is not connected to node } j. \end{cases}$$

Network Terminology

Let us familiarize ourselves with the key terminology of a network [35]:

Actors/Agents/ Point/Node: Social entities such as individuals, cities or organizations.
Arc/Tie/Edge/Link/Line: It represents interactions or relationships among actors.
Dyads: Represented by a pair of actors and the relationship between them.
Triad: A subgroup of three actors and the relationships among them.
Subgroup: Consists of nodes and the relations existing among them.
Group: Set of all actors under observation, on which ties are to be measured.
Relation: A set of a specific kind of ties among members of a group.
Social network: A finite set or sets of actors defined with the relationship or relationships among them.

2.3 Social Network Analysis

Social network analysis (SNA) may be defined as the measuring and mapping of relationships and flows among computers, URLs, people, groups, organizations and other connected knowledge/information entities [40].

2.3.1 Levels of Analysis

Generally, network methods are appropriate at certain levels of analysis [14, 35], which are discussed below:

Actor Level network is analyzed at actor level. The scope of this analysis is the properties and the associated methods pertaining just to actors, e.g. finding the "prominence" of an actor within a group, as measured by metrics such as centrality and prestige, actor-level expansiveness and popularity parameters embedded in stochastic models and metrics for individual roles such as bridges, isolates and liaisons.

Dyadic Level The type of analysis that is applicable to pairs of nodes and the ties between them. It includes finding the distance and reachability between any two nodes, structural and any other notions of equivalence, and probabilistic tendencies towards correlativity and the statistical model-based dyadic analyses for the modelling of various states of a dyad.

Triadic Level This is based on theoretical statements about the balance and transitivity and hypothesizes the behaviours for groups of three actors and their interconnections.

Subset Level This type of analysis helps researchers to study and find subsets of actors that are homogenous with respect to some properties of the network, e.g., cliques and cohesive subgroups containing actors who are "close" to each other, finding subsets of actors having similar behaviour with respect to certain model parameters from some stochastic models and determining the positions of actors through positional analysis etc.

Network Level Last, there are metrics and methods that work on entire groups and all interconnections. It includes graph theoretical measures such as connectedness and diameter, group level measures of centralization, density and prestige. In addition, this type of analysis also focuses on block models and role algebras.

2.3.2 Metrics (Measures) in Social Network Analysis

We need to understand some important properties of social networks, e.g. size, density, degree, reachability, geodesic distance and diameter. These properties may be used in various forms of social network analysis. Hence, we provide a brief discussion of these properties, as mentioned in [32]:

Centrality This metric gives an approximation of the social power of a node based on degree of its connectivity to other nodes in the network. “degree”, “betweenness” and “closeness” are all measures of centrality.

Betweenness Centrality This centrality measure reflects the number of individuals with whom a person is indirectly connecting through their direct links. A vertex occurring on many shortest paths between other vertices has higher betweenness than those that do not.

Closeness The extent to which an individual is close to all other persons in a network (either directly or indirectly). It may be considered a measure of the time taken by the information to spread from a vertex to other vertices in the network, that are reachable from the given vertex. The shortest distance between two nodes may also be known as the “geodesic distance”.

Degree A count of the number of edges incident upon a given vertex gives the measure of degree centrality. It often reflects the immediate risk of node for catching any information or virus, flowing through the network. If the network is directed, then degree centrality can be measured by two separate measures; namely, indegree and outdegree. Indegree represents the count of the number of edges coming to a vertex, whereas outdegree tells us the number of edges going out of a given vertex to other nodes in the network.

Bridge A bridge is an edge that connects two vertices in different components of the graph. The number of components increases on deleting this edge from the network. It is also known as cut-edge or cut arc or an isthmus. An edge may be a bridge only if it is not contained in any cycle.

Prestige The prestige of an actor may be found by the number of connections entering to it. In other words, it indicates the node’s centrality. Different measures of prestige include “Degree prestige”, “status prestige” and “proximity Prestige”. The prestige of an actor increases as it participates in a greater number of ties, but not necessarily when the ties initiate at the node. The prestige cannot be quantified unless the network is directed. The difference between prestige and centrality is analogous to the difference between outdegree and indegree.

Eigenvector Centrality This indicates the importance of a node in the network. Relative scores are assigned to all nodes in the network based on the rule that ties to high-scoring vertices add more to the score of a node compared with the equal connections to low-scoring nodes. “Katz centrality” and Google’s “PageRank” are variants of eigenvector centrality.

Geodesic Distance This can be defined as the length of the shortest path between two particular nodes and thus reflects the minimum number of connections required to connect them.

Reach Two nodes of a network are reachable if a path (and thus a walk) exists between them.

Clustering Coefficient This measures the likelihood that two neighbours of a node are also neighbours of each other. The higher the clustering coefficient, the greater the level of "cliquishness".

Cohesion This may be defined as the extent to which nodes are connected directly to each other through cohesive bonds. Groups are identified as "cliques" if every node is directly connected to every other node, "social circles" if there is less rigidness of direct interaction, or if precision is wanted, then groups are "structurally cohesive blocks".

3 Community Detection in Social Networks

Community detection (CD) in a social network has emerged as a fundamental tool for social network analysis that lays the foundation for several higher level analytics. Although nowadays the automation of community detection has become a favourite topic of research in the field of computational intelligence, it has already been a longstanding topic of research in sociology, as it has been associated with the study of urbanization, social marketing, criminology and various other areas [8]. It has many applications in biology and in physics. The term "community" first appeared in the book entitled "Gemeinschaft und Gesellschaft" published in 1887 [10].

Moreover, with the widespread use of online social networks such as Facebook, the study of the formation and evolution of community structures has gained greater significance. It is important to study community detection as it may prove to have many applications. For example, it may be used for targeted marketing for improved profit or influential targeted campaigning by election candidates. The study of community identification is based on the network structure [8]. It is associated with partitioning a social network into various cohesive subgroups or clusters in such a way that there is a strong cohesion or interconnection among the nodes of a subgroup, as represented by solid lines between nodes in Fig. 1 and weak cohesion among the nodes of different subgroups, represented by dashed lines in the figure. Figure 1 demonstrates three communities existing in the network [22]. The nodes in a community may have dense interaction owing to inter-personal relationships such as friendship and family, or business relationships etc.

As per Newman [26], the task of dividing a network into various communities by partitioning the nodes of a network into "g" number of groups, whereas minimizing the number of intra-group edges is referred to as "graph partitioning".

Many approaches have been used for community detection in social networks. These include the methodologies and the tools from various other disciplines such as physics, biology, sociology, computer science and applied mathematics. However, none of these algorithms solves the purpose and is not fully reliable. Fortunato [15] performed a comprehensive survey of "community detection in graphs" in 2010. The latest advanced review on community detection was carried out by Poonam Bedi and Chhavi Sharma in 2016. Other reviews available in the literature are

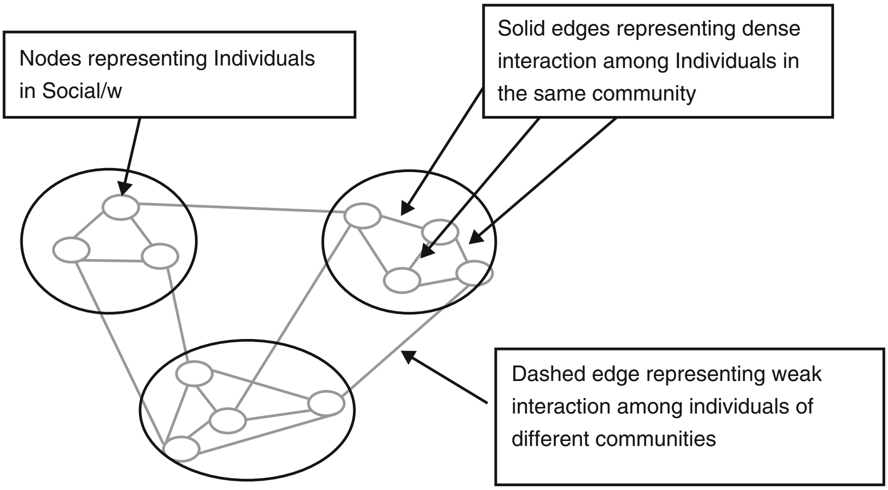

Fig. 1 A visual demonstration of community structure in a network

Lancichinetti and Fortunato [20] in 2010, Danon et al. [31] in 2005, and Khatoon and Aisha Banu [18] in 2015.

3.1 Algorithms for Community Detection

We can broadly categorize the community detection algorithms as follows:

1. **Graph partitioning algorithms:** These are associated with creating some predefined number of partitions of a graph with specific characteristics [32]. A common characteristic to be minimized is the "cut size". A cut is the splitting of the set of the nodes of a graph into two separate subsets and the number of edges between the subset gives the size of the cut. The Kernighan-Lin algorithm is one of the earliest heuristic algorithms for graph partitioning with time complexity as O(n^3).

   This algorithm was designed to assign the "electronic circuits components to circuit boards to reduce the number of connections between boards" [5, 32]. The algorithm is aimed at maximizing Q, the difference between the number of intra-component edges and the number of inter-component edges. Initially, the algorithm partitions the graph into two components of a predefined size, either at random or by using a specific property of the graph. Then, pairs of vertices are swapped until a maximum value of Q is reached. Some of these swaps are done to escape local maxima.
2. **Hierarchical clustering algorithms:** Social networks often exhibit an interesting property of being a hierarchical structure with smaller communities lying under larger communities. This property calls for hierarchical community

detection algorithms that can unveil the hierarchy of communities in a social network. These algorithms are further categorized as:

(a) **Agglomerative hierarchical algorithms**
(b) **Divisive algorithms**

Agglomerative hierarchical algorithms, $O(n^3)$, follow the bottom up approach in which two lower level communities combine or merge to form a higher-level community if they are similar to each other [34]. These algorithms begin by considering each vertex as a community of its own and proceed upward until these are stopped by some stopping criteria, failing which these algorithms end up with the whole graph as a community Fig. 2. On the contrary, Divisive algorithms, as represented in Fig. 3, follow the top-down approach and split communities iteratively into two parts by removing edges between the least similar vertices. It starts with the whole graph as one community and successively

Agglomerative Algorithm 1: A Generic Agglomerative Algorithm

```
input : An input graph G
output: A partition into communities
Compute similarity of every vertex pair;
Put every pair of vertices with corresponding similarity value into a
priority queue Q;
Start with every vertex as a community on its own;
while stopping condition not met do
        if Q is empty then
            break;
        if current partition satisfies the specified requirements then
              break;
Remove the most similar pair of vertices from Q and merge their
corresponding communities into a single community;
          Return the computed partition into communities:
```

Fig. 2 Pseudo code for the hierarchical agglomerative clustering algorithms

Algorithm 2: A generic divisive algorithm

```
input : An input graph G
output: A partition into communities
//Compute similarity of every vertex pair
//Put every pair of vertices with corresponding similarity value into
a priority queue Q//
    1. Start with all vertices as a single community
    2. while stopping condition not yet met do
         if Q is empty then
           break
         if current partition satisfies the specified requirements
       then
             break
    3. Remove the least similar pair of vertices from Q and remove
       the edge between them
Return the computed partition into communities
```

Fig. 3 Pseudo-code for hierarchical divisive algorithm

divides the node set into two subsets until the stopping criterion is met. The most common divisive algorithm is the Girvan–Newman (GN) algorithm [26]. This algorithm depends on the edge-betweenness, i.e. finding the edges that are the most obvious choices for being "between" communities or in other words, finding "inter-community" edges. The edge-betweenness algorithm for sparse graphs takes $O(m^2n)$ and is $O(n^3)$ time, where m is the number of edges, and n denotes the number of nodes. The edge-betweenness algorithm runs faster than random walk betweenness or current-flow [$O(n^2)$ versus $O(n^3)$] on sparse graphs. In addition, the Girvan–Newman algorithm with edge betweenness gives better results in practical applications than those obtained by adopting other centrality measures [41].

Although hierarchical clustering-based algorithms have proved to be better than the graph partition-based clustering algorithms, these algorithms do not perform global optimization of the objective function. Instead, hierarchical clustering techniques perform local optimization of the objective function and decide which of the closest clusters are to be joined (or split in the case of divisive hierarchical clustering) at each step, where "closeness" is specified by specific measures of proximity. The results of these algorithms also depend on the conceptual visualization of clusters and the cluster proximity is taken into account. Moreover, agglomerative hierarchical clustering techniques are adversely affected by the outliers and noise, non-convex shapes and are inclined to break large clusters.

3. **Greedy modularity-based algorithms:** Modularity has been the most widely used quality function for community detection. Several techniques based on modularity use the greedy approach, such as agglomerative hierarchical clustering based on modularity [26]. At the beginning, every vertex is considered to be a separate community, and communities are merged iteratively such that each merge is locally optimal (i.e. yields the largest increase in the current value of modularity). Modularity has been defined as

$$Q = \sum e_{ii} - a_i^2$$

where e_{ii} represents the fraction of the edges that connect nodes in community i, e_{ij} denotes the fraction of the edges that connect nodes in two different communities, i and j, and

$$a_i = \sum_i f_{ij}$$

is the fraction of edges that are incident on the vertices in community i. The value $Q = 1$ indicates a network with a strong community structure.

The algorithm stops when it has reached the maximum modularity, so that it gives a grouping and a dendrogram. However, it is known to suffer from a resolution limit, i.e. communities below a given size threshold (depending on the number of nodes and edges) are always merged with neighbouring communities. Other methods based on greedy techniques include simulated annealing, external optimization and spectral optimization.

4. **Community detection in weighted network:** Community identification in weighted networks uses the greedy approach, which emphasizes maximizing the net weight of all the selected clusters and minimizing the similarity among the selected clusters. Then the calculation of the total weight of the selected clusters and the similarity between them is done [29]. The limitations of this approach are that it assigns each node to exactly one cluster, thus ensuring that every cluster has at least one object. It also ascertains that particular numbers of clusters are

selected, and ensures that a cluster must be selected if a data object is assigned to it [18].

5. **Clustering based on edge betweenness [9]:** This is a hierarchical decomposition methodology where edges are removed in the descending order of their edge-betweenness scores (the number of shortest paths having a particular edge). Although this algorithm gives good results, it is slow owing to the repeated calculation of the edge betweenness score after every edge removal [16]. The modularity may be represented as:

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{(K_i K_j)}{2m}\right]\delta(c_i, c_j),$$

where $m = \frac{1}{2}\sum_{ij} A_{ij}$ represents the number of edges in the graph, while k_i, k_j are degrees of vertices i and j, and $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise.

6. **Community detection based on random walk [9, 10]:** This method is based on "random walks on the graph". The basic idea is to perform a random walk on the graph that tends to stay within the same community because there are very few edges that can take us outside of the given community. The walktrap algorithm runs shorter walks of 3-4-5 steps depending on the parameter used and uses the result of these random walks to agglomeratively merge separate communities to form a larger community. It may be combined with the modularity score to select where to cut the dendrogram. It is a little slower than the fast greedy approach but is more accurate.

7. **Leading eigenvector-based community detection:** In 2006, Newman [27] introduced another approach to characterizing the modularity matrix in terms of eigenvectors and deducing modularity by using the following equation:

$$Q = \frac{1}{4m} s^T B s,$$

where the modularity matrix is given by $B_{ij} = \left[A_{ij} - \frac{K_i K_j}{2m}\right]$ and modularity is given by the eigenvectors of the modularity matrix. The algorithm runs in $O(n^2 \log n)$ time, where $\log n$ represents the average depth of the dendrogram.

8. **Label propagation-based community detection:** Raghavan et al. in 2007 proposed an algorithm to detect community structure in the network based on label propagation that runs in non-linear time and does not require any previous information about communities or optimization of the pre-defined objective function. In this algorithm, a label is provided to each node. It then proceeds iteratively and updates the labels to nodes in a way that each node is assigned the most frequently occurring label among its neighbours [31]. The process stops each node from possessing the most frequently occurring label in its neighbourhood. It is very fast but yields different results based on the initial configuration (which is decided randomly); therefore, the method should be

carried out a many times (say, 1000 times for a graph) and a consensus labelling should be built, which could be tedious.

3.2 Community Detection Algorithms for Overlapping Networks

Although, the field of community detection has witnessed a lot of work, unfortunately, most of the community detection algorithms assume that communities in a network are discrete, which is far from the case. In social networks, communities may be overlapping, i.e. a node may fall into more than one community. In other words, communities in real-life networks are not disjoint. For instance, a person on Facebook may belong to a family group, an old-friends group, a hobby group, a business group and many other groups. Thus, communities with this type of node could hardly be discrete.

On review, disjoint community detection was found to follow five research lines, i.e. modularity maximization, spectral clustering, differential equation, random walks, and statistical mechanics to identify communities. These classic algorithms for community detection may not work well for the real structure of social networks. Overlapping community detection may be categorized into two classes: clique percolation-based [37] and non-clique percolation-based [20, 21]. Palla et al. [28] provided a solution to an overlapping community detection problem based on a clique percolation [20, 21]. The algorithm first finds all cliques in the network and identifies communities using component analysis of a clique–clique overlap matrix [6]. The clique percolation method runs in $O(\exp(n))$ time. Non-clique-based methods include label propagation, random walk [6], link partitioning and line graph, fuzzy detection, local expansion and optimization, agent-based and dynamical algorithms [20, 21].

3.3 Community Detection Algorithms for Dynamic Networks

A network is static if its configuration does not change with time. A network that keeps changing its configuration is dynamic. Social networks are characterized by their dynamic nature as the membership of a node evolves and keeps changing over time. Thus, the dynamics of the network change over time. As per Fortunato [15] and Deepjyoti Choudhury [10], three algorithms have been provided for dynamic algorithm: the spin model, random walk and synchronization. Wolf et al. proposed a mathematical and computational model for the analysis of dynamic communities that was based on interactions among nodes in the network [6].

4 Nature Inspired Algorithms and Their Applications in Community Detection

Computational intelligence as shown in Fig. 4 refers to a set of evolutionary techniques and methods of addressing complicated real world problems for which conventional mathematical tools and techniques are ineffective. The methods used in computational intelligence are close to the human way of reasoning and uses combinations of the following main techniques:

Nature-inspired algorithm (NIA) has evolved as a promising solution for the optimization of hard problems in a reasonable manner, and thus has found a niche among researchers tackling hard real-world problems, i.e. the problems for which exact solutions do not exist, but for which an approximate solution may be found.

Nature-inspired algorithms are inspired by the various processes observed from nature. The motive of developing such computing methods is to optimize engineering problems such as numerical benchmark functions, multi-objective functions and solving NP-hard problems for higher dimensions and greater sizes of problem.

Nature-inspired computation may be categorized as in Fig. 5.

There has been many solutions to community detection in social networks that comes from the family of nature-inspired algorithms, i.e. genetic algorithm,

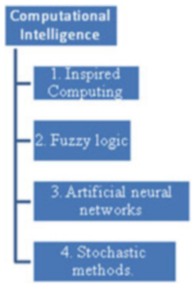

Fig. 4 Evolutionary techniques under computational intelligence

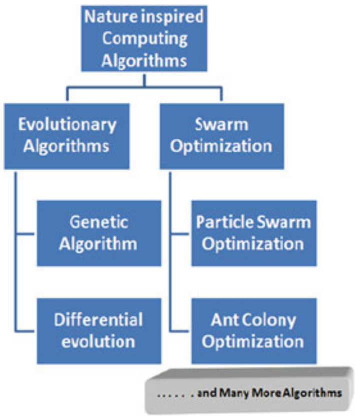

Fig. 5 Categories of nature-inspired computing algorithms

artificial bee colony optimization, ant lion optimization, hybrid of improved teacher learner based optimization (I-TLBO) and group search optimization (GSO) algorithms) [11] and multi-objective evolutionary algorithm etc. Thus, there is a good probability of nature-inspired algorithms tackling community detection problems in social networks efficiently.

The functioning of nature-inspired algorithms may be generalized in the following way to optimize any problem solution.

General Steps in Solving a Problem Using Nature-Inspired Algorithms [13]

1. Initialize population or search agents
2. Find the fitness function
3. Iterate until the stopping criteria are achieved or the maximum iterations are completed
    a. Calculate the fitness function for all search agents
    b. Find the best optimal fitness value
    c. Evaluate the new fitness value obtained against the previous one
    d. If the new fitness value is better than the previous value replace the fitness value with the newly obtained value
    e. Return the fitness value
4. Exit.

In this section we explore the application of various nature inspired algorithms in community detection.

Community detection methods have been proposed under two categories:

1. Community detection for unsigned networks
2. Community detection for signed networks

Communities of unsigned social networks can be identified as the group of nodes with more intense interconnections within the group and less intense interconnections among different communities. In recent years, a new type of social network has received a high level of attention from researchers across the globe, that is, the signed social network. In signed social networks, community structure is defined not only by the extent of interlinking but also by the signs of the links between two nodes. There are more positive signs between nodes in a community and more negative signs between nodes of two different communities. Positive edges represent the like, trust or affinity between two nodes or individuals, whereas negative edges represent the dislike, distrust or aversion between two nodes. Dataset for signed social networks may be found at Slashdot news review site, Wikipedia vote site and Epinions consumer review site [2].

4.1 Community Detection for Static, Unsigned and Non-overlapping Networks

Several evolutionary algorithms have been mapped to provide optimized solutions to community detection problems for static, unsigned and non-overlapping social networks. The following section provides a description of the applications of some important evolutionary algorithms.

4.1.1 A Nature-Inspired Metaheuristic Lion Optimization Algorithm

The lion optimization algorithm (LOA) was proposed by Maziar Yazdani and Fariborz Jolai in 2015. LOA is metaheuristic algorithm that falls under stochastic optimization and is characterized by the generation of different solutions for the problem in each run. The LOA is inspired by simulation of the special style exhibited by lions for capturing prey, territorial marking, migration, difference between the life style of nomad and resident lions. Lions thus have two types of social behaviour, nomadic and resident, and may switch the type of social organization. They live in groups called prides. The territory is the area in which a pride lives. The lions and the cubs protect the pride from nomadic lions. The LOA finds the optimal solution based on a lion's behaviour regarding territory defence and territory takeover. Through territory defence, the LOA evaluates the existing solution (territorial lion) and newly generated solution (nomadic lion) and the best one becomes the new solution, replacing the previous one. By territorial takeover, selection of the best territorial male is made from among the old territorial and new territorial males.

In 2015, Ramadan Babers et al. proposed a nature-inspired meta-heuristic LOA for community detection [2]. It used a normalized mutual index and modularity as the performance functions. Normalized mutual information (NMI), introduced by A. Lancichinetti, is a quality function and has become a standard metric for the evaluation of community detection algorithms. NMI is useful only when communities of vertices are known.

Given two partitions A and B of a network in communities, let C be the confusion matrix, whose element Cij is the number of nodes of the community i of partition A that are also in the community j of partition B. The normalized mutual information $I\,(A; B)$ is defined as:

$$\mathrm{NMI}(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} c_{ij}\log\left(\frac{c_{ij}N}{c_{i.}c_{.j}}\right)}{\sum_{i=1}^{C_A} .C_{i.}\log\left(\frac{c_i}{N}\right) + \sum_{j=1}^{C_B} c_{.j}\log\left(\frac{c_j}{N}\right)}$$

where $C_A(C_B)$ is the number of groups in partition A (B), $C_i.(c_{(.j)})$ is the sum of the elements in row i (column j) and N is the number of nodes.

The experiments were performed on the Zachary Karate Club, the Bottlenose Dolphin Network, and the American College Football Network datasets. The results showed that the LOA seemed promising in terms of accuracy and quite successfully found optimized community structures for the presented datasets. The results obtained are summarized later in Table 1.

Table 1 Comparative analysis of evolutionary algorithm

(a) Modularity obtained by various algorithms

Reference	Methods	*Modularity*		
		Karate	*Football*	*Dolphins*
Proposed	GWO	0.4145	0.6038	0.5222
[16]	GN	0.38	0.577	0.495
[3]	TL-GSO	0.4185	0.6022	0.5019
[3]	GSO-1	0.3845	0.4276	0.4298
[3]	CNM	0.3806	0.5497	0.4954
[3]	Multilevel	0.4188	0.6045	0.5185
[11]	Discrete bat algorithm	0.4696	0.612	0.5498
[2, 38]	Lion optimization	0.46	0.89	0.59
[30]	MOGA-Net	0.4159	0.515	0.505
[1]	Bi-objective community detection	0.419	0.577	0.507
[3, 36]	Memetic algorithm	0.402	0.6044	0.5232

(b) NMI index obtained by various algorithms

Reference	Methods	*NMI*		
		Karate	*Football*	*Dolphins*
Proposed	GWO	0.8895	0.9734	0.8629
[16]	GN	0.692	0.762	0.573
[11]	Discrete bat algorithm	0.6873	0.84786	0.5867
[2, 38]	Lion optimization	0.67	0.91	0.58
[30]	MOGA-Net	1	0.796	1
[39]	MPCA	1	0.923	0.956
[30]	GA	0.818	0.805	0.935
[1]	Bi-objective community detect	0.695	0.878	0.615
[3, 36]	Memetic algorithm	0.693	0.762	0.573

(c) Number of communities identified by various algorithms

Reference	Methods	*No. of communities*		
		Karate	*Football*	*Dolphins*
Proposed	GWO	4	10	5
[3]	TL-GSO	4	10	5
[3]	GSO-1	3	7	5
[3]	CNM	3	6	4
[3]	Multilevel	4	10	5
[2, 38]	Lion optimization	4	10	5
[3, 36]	Memetic algorithm	3	7	4

4.1.2 TL-GSO: A Hybrid Approach to Mine Communities from Social Networks

The TL-GSO, proposed by Hema Banati and Nidhi Arora in 2015, is a hybrid algorithm based on the two latest evolutionary algorithms, the group search optimization (GSO) algorithm and the improved teacher learner-based optimization algorithm (I-TLBO). TL-GSO employs a single-point crossover and works on the same basis as GSO's Producer-Scrounger (PS) searching model, improving the earlier algorithm by employing multiple producers per generation instead of one, thus resulting in faster convergence of the best overall solution. TL-GSO was executed on four real-world data sets, i.e. the Karate Club Network, the Dolphin Network, the American Football Network and the Political Books Network. The results obtained from the algorithm have been summarized in Table 1.

4.1.3 Multi-Objective Genetic Algorithm

Pizzuti proposed a new algorithm for solve the problem by using a genetic algorithm. The algorithm uses the density measure and a latest concept of community score as a global measure to segment the network in clusters. The algorithm is aimed at maximizing this score [39]. The multi-objective genetic algorithm (MOGA-Net) is basically a non-dominated sorting genetic algorithm (NSGA-II) that builds and ranks the individual's population on a non-dominance basis [19]. The algorithm produces network communities at different levels of hierarchies, thus using a hierarchical clustering technique and automatically resolute communities are formed [24]. The author has used two competing objectives. The first one is about maximizing the interaction among the nodes in the same cluster and the second objective is to minimize the inert community interconnections. A multi-objective optimization problem $(\Omega, F_1, F_2, \dots, F_t)$ is defined as

$$\min F_i(S), i = 1, \dots, t \text{ subject to } S \epsilon \Omega$$

where $\Omega = \{S_1, S_2 \dots, S_k\}$ is the set of plausible clustering of a network, and $F = \{F_1, F_2, \dots, F_t\}$ is a set of t single criterion functions. Each $F_i : \Omega \rightarrow R$ is a distinct objective function for determining the feasibility of the clustering thus obtained.

The datasets used to test the algorithm are the Zachary Karate Club, the Bottlenose Dolphin Network, the American College Football Network and Krebs' Book. The pro side of the MOGA-Net algorithm is that it provides a set of solutions at different levels of the hierarchy by giving the opportunity to analyze the network structure at different resolution levels and hence proves to be better than the single objective approaches.

4.1.4 Multi-Population Cultural Algorithm

A cultural algorithm is an evolutionary framework based on knowledge and employs knowledge in the evolution of optimal solutions. Devised by Pooya Moradian Zadeh and Ziad Kobti in 2015 [39], the multi-population cultural algorithm (MPCA) is an algorithm in which population space representing the solution is randomly created based on the state space of the network. From among the initial solutions a belief space consisting of the solutions with better fitness values is generated. In each generation, the set of the best individuals of each population is used to update the belief space. The algorithm is evaluated against GA-Net, MOGA-Net, DECD (differential evolution) and the NMI value has been used as the performance evaluation function to measure the similarity between the newly detected community structure and the actual one. The experiment was performed on the synthetic network and on the real datasets of the Zachary Karate Club, Dolphin and American Football Networks. The results have been summarized in Table 1.

Other algorithms that have been applied for the community detection of unsigned, static and non-overlapping community structure in social networks include bi-objective community detection using the genetic algorithm proposed by Rohan Agrawal in 2011 [1], the GA-Net algorithm [30] proposed by Clara Pizzuti in 2008, the memetic algorithm for community detection in networks proposed by Maoguo Gong et al. in 2011 [17] and the multi-objective community detection based on memetic algorithm [36] proposed by Peng Wu and Li Pan in 2015.

4.2 Community Detection for Signed Networks

This section describes the applications of various evolutionary algorithms to signed social networks.

4.2.1 Cultural Algorithm Based on Artificial Bee Colony Optimization

The cultural algorithm based on artificial bee colony (ABC) optimization was proposed by HU Baofang in 2015. The experiment was conducted on synthetic signed networks generated with three different sizes [4] and two real-world networks (the Gahuku-Gama Sub tribes [GGS] network, the Slovene Parliamentary Party (SPP) network). The cultural algorithm is a knowledge- and belief-based evolutionary framework that employs knowledge to accelerate the evolution process and conduct the direction of evolution. Topological knowledge is employed in our proposed algorithm, which is very suitable for local searching for employees and on-looker bees. The objective function used for the experiment is extended modularity in signed networks. A single objective optimization problem (Ω; F) is used, as shown in the equation below.

$$\mathbf{Min}\ F(S), \mathbf{s.t}\ S \in \Omega$$

where $F(S)$ is an objective function to be optimized, and $\Omega = \{S1, S2, \ldots, Sm\}$ is the set of possible communities of a signed network. Fitness function was given by:

$$F(S) = \sum_{ij} \lambda A_{ij}^{-} \delta(c_i, c_j) + (1 - \lambda) A_{ij}^{+} (1 - \delta(c_i, c_j)$$

where A_{ij} represents the weight of the edge between vertices i and j. A_{ij}^{-} denotes the negative edges between i and j and A_{ij}^{+} is the positive edges between i and j. Here, c_i is the community number of nodes i and $\delta(c_i, c_j) = 1$, if $\sigma_i = \sigma_j$, else $\delta(c_i, c_j) = 0.\lambda$ is a parameter through which the two types of edges, positive and negative, can be balanced and $0 < \lambda \leq 1$. The performance evaluation of the algorithm is based on the NMI metric. The algorithm was compared with the ABC algorithm, GA-Net and MPCA for signed networks and the author showed that the cultural algorithm proved to be better than those algorithms.

4.2.2 Multi-Objective Evolutionary Algorithm Based on Similarity

The multi-objective evolutionary algorithm (MEA-SN) was proposed by Chenlong Liu, Jing Liu and Zhongzhou Jiang in 2014. The authors considered both the link density and the sign of the network. Two objectives are proposed. The first objective is to bring all positive links in communities, whereas the second is to keep all negative links as inter-community links. As the MEAs-SN can switch between different representations during the evolutionary process, it thus benefits from both representations and can detect both separate and overlapping communities [23]. The algorithm makes use of a modified modularity function for signed networks, represented by Q_{signed} and modified modularity function for overlapping community structures, represented by Q_{ov}. The authors generated synthetic networks using the Lancichinetti–Fortunato–Radicchi (LFR) benchmark, suitable for both separate and overlapping networks. The results are evaluated using NMI and modularity values, and are shown in Table 1.

4.2.3 Discrete Bat Algorithm

Inspired by the bat algorithm proposed by Xin-She Yang, Wang Chunyu proposed a novel discrete bat algorithm [11] in 2015. The authors of this algorithm adopted an ordered adjacency list as the encoded form, and used modular Q function as the objective function. The accuracy of division, i.e. the rate of correct partitioning, is used as the evaluation standard, and simulation is done by using MATLAB R2013a. The analysis of simulation results shows that the discrete bat algorithm is effective and efficient at detecting communities in complex networks. The results are presented in Table 1.

4.3 Community Detection for Overlapping Communities

4.3.1 Genetic Algorithm for Identifying Overlapping Communities in Social Networks

Brian Dickinson et al. proposed a genetic algorithm in 2013 for the identification of overlapping communities in social networks using an optimized search space [12]. The author presents two algorithms for community detection. One is the LabelRank algorithm, which, because of its deterministic nature, is restricted to very few candidate solutions. The other algorithm is the genetic algorithm, which employs a restricted edge-based clustering technique to identify overlapping communities (OGA) by maximizing a modified and efficient overlapping modularity function. The experiments were performed on several real-world datasets, including the Karate Network, the Pilgrim Network, Lesmis Network, Dolphin Network and Football Network. The results are compared against the speaker-listener label propagation algorithm (SLPA). The algorithms are analyzed using the modified Q function and the F-score.

4.3.2 A Dynamic Algorithm for Local Community Detection in Graphs

A dynamic algorithm for local community detection in graphs was proposed by Anita Zakrzewska and David A. Bader in 2015 [40]. It was based on the dynamic seed set expansion, which helps to incrementally update the community when the associated graph changes. The algorithm has severe shortcomings that the community may split apart, but the algorithm is not able to identify this as the community score remains unchanged. The performance improvement is greatest in cases of low latency updates. The dynamic method is faster than re-computation, the experiment was conducted on the Slashdot graph, and precision and recall measures are used for performance evaluation. All these algorithms have been summarized in tabular form in the Appendix.

5 A Novel Approach to Community Detection Using Grey Wolf Optimization

In this section, we first briefly describe a novel algorithm for optimization, i.e. grey wolf optimization (GWO) proposed by Syed Mirzalili et al. in 2014 [25]. This is followed by a novel solution to community detection that uses GWO as a tool for optimization.

5.1 Grey wolf optimization

The GWO algorithm is inspired by the leadership hierarchy and hunting mechanism of grey wolves found in the mountains, forests and plains of North America, Europe and Asia. Grey wolves are characterized by a bushy tail and powerful teeth. They live and hunt in groups called packs. The average group size is 5–12. They exhibit an attractive social organization, with four categories of wolves, i.e. alpha, beta, delta and omega, and there is a leadership hierarchy in the pack.

(a) α is the dominant leader and decision maker in the pack.
(b) β & δ—the betas are subordinate to the alpha wolf and help the alpha in decision-making. The beta wolf is probably the next best candidate to be the alpha and can be a male or a female. Delta wolves must follow the alpha and betas, but they dominate the omegas.
(c) Ω-Omegas are the lowest ranking wolves in the pack. They are the scapegoats in the pack and always have to submit to the alphas, betas and deltas.

Apart from exhibiting the social hierarchy, they also exhibit other important social behaviour of particular interest, such as group hunting. The grey wolves perform group hunting, which may be divided into the following phases [25]:

1. Tracking, chasing and approaching the prey
2. Pursuing, encircling and harassing the prey until it stops moving
3. Attacking the prey.

The mathematical model of the encircling behaviour is represented by the equations:
equations

$$\mathbf{D} = |\mathbf{C}\mathbf{X}_p - \mathbf{A}\mathbf{X}(t)| \tag{1}$$

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - AD \tag{2}$$

where X_p is the position vector of the prey, X indicates the position vector of a grey wolf and A and C are coefficient vectors given by

$$\mathbf{A} = \mathbf{2ar}_1 - \mathbf{a} \tag{3}$$

$$\mathbf{C} = \mathbf{2r}_2 \tag{4}$$

where,
T is the current iteration
X is the position vector of a wolf, and
r1 and r2 are random vectors ϵ [0,1] and **a** vary linearly from 2 to 0.

5.1.1 The Mathematical Model for Hunting

The hunting mechanism of grey wolves may be modelled by the following equations:

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha(\mathbf{t}) - \mathbf{X}(\mathbf{t})|, \mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta(\mathbf{t}) - \mathbf{X}(\mathbf{t})|, \mathbf{D}_\delta = |\mathbf{C}_3 \cdot \mathbf{X}_\delta(\mathbf{t}) - \mathbf{X}(\mathbf{t})| \tag{5}$$

$$\mathbf{X}_1 = \mathbf{X}_\alpha(\mathbf{t}) - \mathbf{A}_1.(\mathbf{D}_\alpha), \mathbf{X}_2 = \mathbf{X}_\beta(\mathbf{t}) - \mathbf{A}_2.(\mathbf{D}\boldsymbol{\beta}), \mathbf{X}_3 = \mathbf{X}_\delta(\mathbf{t}) - \mathbf{A}_3.(\mathbf{D}_\delta) \tag{6}$$

$$\mathbf{X}(\mathbf{t}+\mathbf{1}) = \frac{(\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3)}{3} \tag{7}$$

where t indicates the current iteration, $X_\alpha(t)$, $X_\beta(t)$ and $X_\delta(t)$ are the position of the grey wolf α, β and δ at tth iteration, $X(t+1)$ presents the position of the grey wolf at $(t+1)$th iteration.

$$\mathbf{A(.)} = 2 \cdot a \cdot \mathrm{rand}(0, 1) - a \tag{8}$$

$$\mathbf{C(.)} = 2 \cdot \mathrm{rand}(0, 1) \tag{9}$$

Where "a" is the linear value that varies from 2 to 0 according to iteration. A(.) and C(.) are the coefficient vectors of the α, β and δ wolves. The GWO algorithm allows its search agents to update their position based on the location of α, β and δ and attack the prey. Although the proposed encircling mechanism demonstrates exploration to some extent, it is prone to stagnation in local solutions with these operators and hence needs more operators for better exploration.

5.2 Community Detection Using Grey Wolf Optimization

The pseudo code for the implementation of the GWO algorithm for community detection is presented below.

Pseudo code for the GWO algorithm

1. Initialize the alpha, beta and delta positions
2. Initialize the position of the search agents
3. Loop until maximum iteration is not reached:
   (a) Calculate the objective function for each search agent
   (b) Update alpha, beta, delta and omega positions

(continued)

   (c) Evaluate the new fitness value obtained against the previous one
   (d) Return the search agents that go beyond the boundaries of the search space.
4. Return the solutions.

We have used the modularity function as the objective function and have used NMI as the performance evaluation criteria together with the modularity of the network.

5.3 *Experiment*

The GWO algorithm was used for community detection for the first time. The experiment was carried out on Intel(®) core(™), i5 CPU @ 2.53 GHz, 4 GB RAM using MATLAB R2010a. The GWO algorithm was used to optimize the modularity function, and uses NMI index values and modularity values as the performance evaluation criteria.

Datasets We used three real-world datasets for the purposes of community detection.

1. The Zachary Karate Club Network, consisting of 34 members represented by the vertices and the relationships among the club president and the karate instructor represented by edges. It simulates the relationships between the club president and the karate instructor and contains 78 edges [2].
2. The Bottlenose Dolphin Network, simulating the behaviour of bottlenose dolphins seen over 7 years. It contains 62 nodes and 318 edges.
3. American College Football Network [2] demonstrates the football games between American Colleges during a regular season in fall 2000. It contains 115 nodes and 1226 edges.

Parameter Initialization

```
 No. of search agents : 50
Maximum no. of iteration : 50
No. of trial runs : 10
Lower bound =0, upper bound = 10 (for the Karate Network)
Lower bound =0, upper bound = 20 (for the Football Network)
Lower bound =0, upper bound = 10 (for the Dolphin Network)
```

Experimental Results The GWO algorithm was first proposed for community detection and was implemented on three different real-world datasets for the extraction of community structure in those real networks. The GWO algorithm was executed for ten trial runs for each dataset. The modularity and NMI values obtained across the ten trial runs were averaged to provide the final modularity and NMI values, as shown in Table 1a and b respectively.

The results of modularity value and NMI values thus obtained from the experimental setup are further compared with the state of the art algorithms that were employed earlier for community detection. These include TL-GSO, the discrete bat algorithm, lion optimization, MOGA-Net etc.

For better analysis, we compare our results of average modularity and average NMI with those algorithms that have also employed similar criteria. The source of modularity values obtained from various algorithms is mentioned in the first column. The results are summarized under Table 1a for modularity value, Table 1b for comparative analysis of NMI values and Table 1c for the number of communities detected by various algorithms.

Figure 6 shows that the GWO algorithm demonstrates competitive modularity compared with the GN, TL-GSO, lion optimization and memetic algorithms, bi-objective community detection (BOCD) and others, with discrete bat algorithm and lion optimization algorithms as the exceptions.

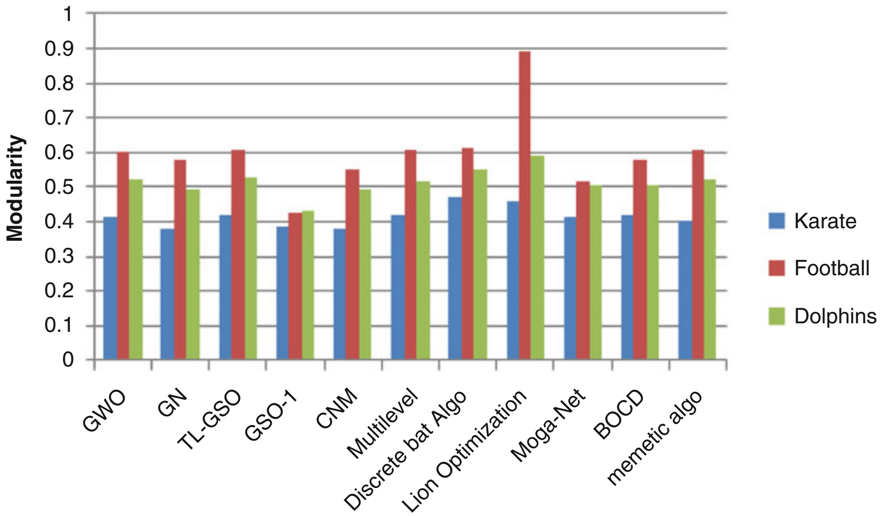

Fig. 6 Modularity comparison

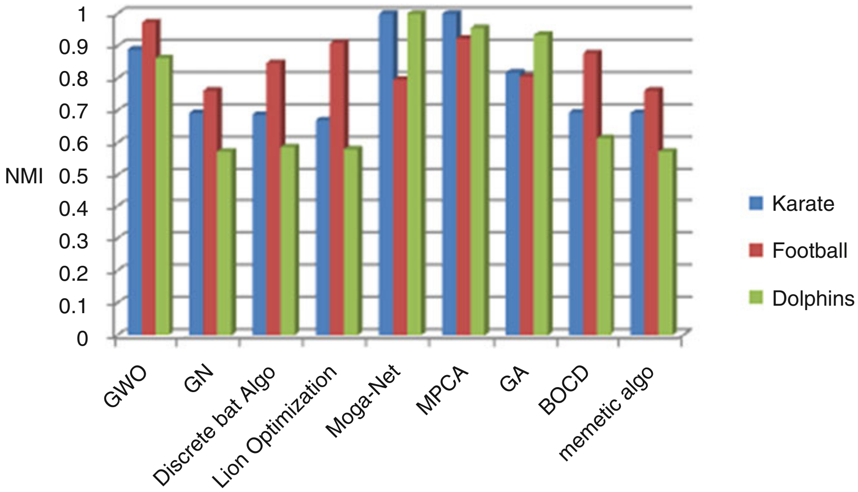

Fig. 7 Normalized mutual information comparative analysis

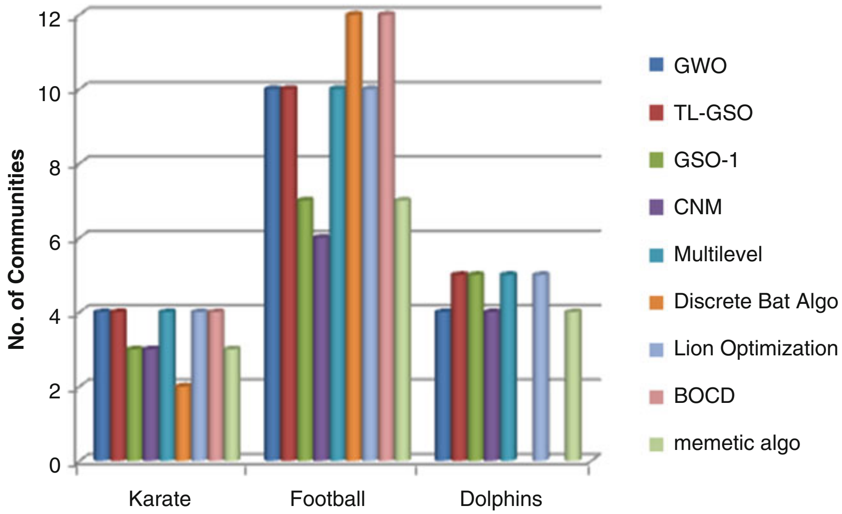

Fig. 8 Variation of modularity value across ten trials

Figure 7 demonstrates the comparison of NMI values obtained through the GWO algorithm with lion optimization, discrete bat algorithm and TL-GSO algorithms, MOGA-Net and other contemporary evolutionary algorithms.

Figure 8 demonstrates the number of communities identified by various evolutionary algorithms for the three different datasets, compared with our proposed algorithm.

Table 2 Experimental results for the GWO algorithm

Dataset	Modularity			NMI		
	Highest	Lowest	Average	Highest	Lowest	Average
Karate	0.4198	0.3942	0.4145	1	0.815	0.908
Football	0.6046	0.5976	0.6038	1	0.9199	0.9734
Dolphin	0.5277	0.5188	0.5222	1	0.7224	0.863

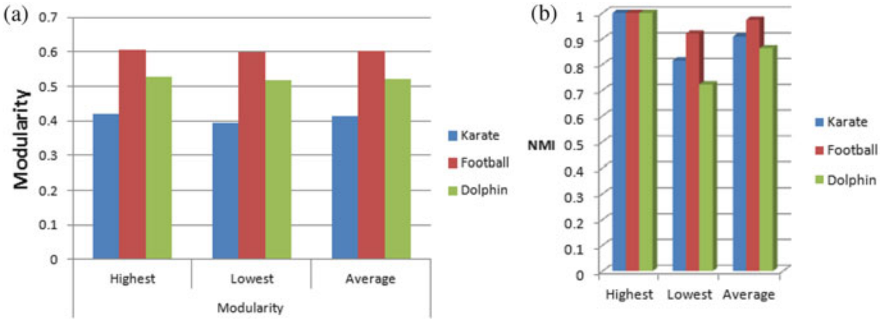

Fig. 9 Experimental results for community detection using the GWO algorithm. (**a**) Modularity results for GWO. (**b**) NMI results for the GWO algorithm

The lowest, highest and average values of modularity and NMI obtained through ten runs of the GWO algorithm for each dataset have been demonstrated in Table 2 and can be modelled by Fig. 9a and b respectively.

6 Conclusion

The experiments were performed on the three different datasets, i.e. the Karate Club Network (with 34 nodes and 78 edges), the Bottlenose Dolphin Network (with 62 nodes and 159 edges) and the American College Football Network(with 115 nodes and 613 edges) using the GWO algorithm for 10 runs. The highest, lowest and average values for the modularity and the NMI thus obtained have been summarized in Table 2. The highest modularity value has been reported to be 0.4198 for the Karate Club dataset, 0.6046 for the Football dataset and 0.5277 for the Dolphin dataset respectively, which is on a par with other evolutionary algorithms. The GWO algorithm resulted in four communities for the Karate Club dataset, ten communities for the Football dataset and five communities for the Dolphin dataset.

The results demonstrate the promising nature of the GWO algorithm. The algorithm does not need the number of communities of the dataset in advance. It optimizes the objective function at each iteration to finally converge at the final result. The algorithm has a higher rate of convergence. It converged in less than 50 iterations compared with other nature-inspired algorithms that took 100 s of iterations to converge.

Hence, the GWO algorithm is capable of generating the optimized results in a shorter period. The algorithm results in higher values of modularity than other evolutionary algorithms. The NMI values obtained from the GWO algorithm almost approach 1, which shows the accuracy of the results.

Appendix

See Fig. 10.

S.No	Algorithm	Author / Authors	Year	Performance evaluation parameters	Community Detection			Data sets
					Signed / Unsigned	Static / Dynamic	Overlapping/ Non-overlapping	
1	ABC based cultural algorithm	HU Baofang	2015	NMI	Signed	Static	Non-overlapping	artificial signed networks generated with 3 different sizes, and two real world networks (Slovene Parliamentary Party network (Spp), the Gahuku-Gama Subtribes netw3.ork (GGS)).
2	LION optimization algorithm	Ramadan Babera*, Aboul Ella, Hassanien, and Neveen I. Ghali	2015	NMI, community fitness, modularity	Unsigned	Static	Non-overlapping	Zachary kate Club, Bottlenose Dolphin network, American college Football Network
3	MEA-SN	Chenlong Liu, Jing Liu, Zhongzhou Jiang	2014	NMI, modularity	both signed and unsigned	Static	both overlapping and non-Overlapping	Synthetic networks are generated of size 1000, 5000, and 10000 nodes using (LFR) benchmark Real network- 1. Slovene parliamentary party Network, 2. Gahuku-Gama Subtribes network, 3. Karate Club network 4. Dolphin network, 5. Football Network
4	Discrete Bat Algorithm,	Wang chuyu* and Pan Yun	2015	accuracy of division, Modularity	unsigned	Static		Zachary Kate Club
5	GWO	Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis	2014	NMI, Modularity function value	Unsigned	Static	Non Overlapping	Zachary Kate Club, Bottlenose Dolphin network, American college Football Network
6	TL-GSO	Hema Banati, Nidhi Arora	2015	NMI, Modularity	Unsigned	Static	Non Overlapping	Karate Club, Dolphin, Football, Political Books
7	Multiobjective Based Genetic Algorithm (MOGA-Net)	Clara Pizzuti	2009	Modularity and NMI	Unsigned	Static	Non Overlapping	Karate Club, Dolphin, Football, Krebs' Book
8	Multi Population Cultural Algorithm (MPCA)	Pooya Moradian Zadeh, Ziad Kobti	2015	NMI	Unsigned	Static	Non Overlapping	Karate Club, Dolphin, Football
9	A Genetic Algorithm for Identifying Overlapping communities (OGA)	Brian Dickinson, Benjamin Valyou, Wei Hu	2013	Modified modularity (Q), F-Score	Unsigned	Static	Overlapping	Karate Network, Pilgrim Network, Lermis, Dolphin and Football Network.
10	A Dynamic Algorithm for Local Community detection in graphs	Anita Zakrzewaka and David A. Bader	2015	Precision and recall	Unsigned	Static	Overlapping	slashdot graph

Fig. 10 Summary of evolutionary algorithms in Community detection

References

1. Agrawal, R.: Bi-objective community detection (BOCD) in networks using genetic algorithm. In: Communications in Computer and Information Science, vol. 168, September 2011. doi:10.1007/978-3-642-22606-9_5. Source: arXiv
2. Babers, R., et al.: A nature-inspired metaheuristic lion optimization algorithm for community detection. In: Computer Engineering Conference (ICENCO), 2015 11th International, IEEE (2015). 978-1-5090-0275-7/15/2015
3. Banati, H., Arora, N.: TL-GSO - a hybrid approach to mine communities from social network. In: 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)
4. Baofang, H.U.: A cultural algorithm based on artificial bee colony optimization for community detection in signed social networks. In: 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (2015)
5. Barbara, D.: An Introduction to Cluster Analysis for Data Mining (2000). Retrieved November 12, 2003, from http://www-users.cs.umn.edu/~han/dmclass/cluster_survey_10_02_00.pdf
6. Bedi, P., Sharma, C.: Community detection in social networks. Adv. Rev. Published online: 19 Feb 2016. doi:10.1002/widm.1178. WIREs Data Mining Knowledge Discovery. doi:10.1002/widm.1178
7. Bhawsar, Y., Thakur, G.S.: Community detection in social networking. J. Inf. Eng. Appl. **3**(6) (2013). www.iiste.org. ISSN: 2224-5782 (print). ISSN: 2225-0506 (online) - Selected from International Conference on Recent Trends in Applied Sciences with Engineering Applications
8. Chen, W.: Community detection in social networks through community formation games. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (2011)
9. Chen, F., Li, K.: Detecting hierarchical structure of community members in social networks. Knowl.-Based Syst. **87**, 3–15 (2015)
10. Choudhury, D., Paul, A.: Community detection in social networks: an overview. Int. J. Res. Eng. Technol. eISSN: 2319-1163. pISSN: 2321-7308
11. Chunyu, W., Yun, P.: Discrete bat algorithm and application in community detection. Open Cybernet. Syst. J. **9**, 967–972 (2015)
12. Dickinson, B.: A genetic algorithm for identifying overlapping communities in social networks using an optimized search space. Soc. Netw. **2**, 193–201 (2013). http://dx.doi.org/10.4236/sn.2013.24019. Published Online October 2013 http://www.scirp.org/journal/sn
13. Dixit, M., et al.: An exhaustive survey on nature inspired optimization algorithms. Int. J. Softw. Eng. Appl. **9**(4), 91–104 (2015). http://dx.doi.org/10.14257/ijseia.2015.9.4.11
14. Eindride, E.: Community Detection in Social Networks, a thesis. Department of Informatics, University of Bergen (2015)
15. Fortunato, S.: Community detection in graphs. arXiv:0906.0612v2 [physics.soc-ph] 25 Jan 2010
16. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002)
17. Gong, M.: Memetic algorithm for community detection in networks. Phys. Rev. E **84**, 056101 (2011)
18. Khatoon, M., Aisha Banu, W.: A survey on community detection methods in social networks. Int. J. Educ. Manage. Eng. **1**, 8–18 (2015)
19. Kim, K., et al.: Multiobjective evolutionary algorithms for dynamic social network clustering. In: GECCO 1́0 Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 1179–1186
20. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. arXiv:0908.1062v2[physics.soc-ph]16 Sep 2010
21. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. PLoS One **6**(4), e18961 (2011). doi:10.1371/journal.pone.0018961

22. Li, K., Pang, Y.: A unified community detection algorithm in complex network. Neurocomputing **130**, 36–43 (2014). www.elsevier.com/locate/neucom
23. Liu, C.: A Multiobjective Evolutionary algorithm based on similarity for community detection from signed social networks. IEEE Trans. Cybernet. **44**(12) (2014)
24. Mahajan, A., Kaur, M.: Various approaches of community detection in complex networks: a glance. Int. J. Inf. Technol. Comput. Sci. **4**, 35–41 (2016). Published Online April 2016 in MECS. http://www.mecs-press.org/DOI:10.5815/ijitcs.2016.04.05
25. Mirjalili, S.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
26. Newman, M.E.J.: Detecting community structure in networks. Eur. Phys. J. B **38**, 321 (2004). doi:10.1140/epjb/e2004-00124-y
27. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**, 036104 (2006)
28. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814–818 (2005)
29. Pizzuti, C.: A multi-objective genetic algorithm for community detection in networks. In: ICTAI 2009, 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, NJ, 2–4 November 2009
30. Pizzuti, C.: A multiobjective genetic algorithm to find communities in complex networks. IEEE Trans. Evol. Comput. **6**(3), 418–430 (2012)
31. Raghavan, U.N., et al.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**, 036106 (2007)
32. Social Network Analysis: Theory and Applications. https://www.politaktiv.org/documents/10157/29141/SocNet_TheoryApp.pdf
33. Steinhaeuser, K., Chawla, N.V.: Community detection in a large real-world social network. In: Social Computing, Behavioral Modeling, and Prediction, pp. 168–175
34. Wang, M., et al.: Community detection in social networks: an indepth benchmarking study with a procedure oriented framework. In: Proceedings of the VLDB Endowment VLDB Endowment Homepage archive, vol. 8(10), June 2015, pp. 998–1009
35. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press, Cambridge (1994)
36. Wu, P., Pan, L.: Multi-objective community detection based on memetic algorithm. PLoS One **10**(5), e0126845. doi:10.1371/journal. pone.0126845 (2015)
37. Xu, Y.: Finding overlapping community from social networks based on community forest model. Knowl.-Based Syst. 1–18 (2016). ISSN: 0950-7051 000
38. Yazdani, M., Jolai, F.: Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J. Comput. Des. Eng. **3**, 24–36 (2016)
39. Zadeh, P.M., Kobti, Z.: A multi-population cultural algorithm for community detection in social networks. In: The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015), Procedia Computer Science, vol. 52, pp. 342–349 (2015)
40. Zakrzewska, A., Bader, D.A.: A dynamic algorithm for local community detection in graphs. In: 2015 Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, ASONAM 2015, Paris, 25–28 August 2015, pp. 559–564
41. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **9**(2), 026113 (2004)