Neural Computing and Applications (2020) 32:9649–9665
https://doi.org/10.1007/s00521-019-04487-0

ORIGINAL ARTICLE

# Genetic algorithm-based community detection in large-scale social networks

Ranjan Kumar Behera[1] · Debadatta Naik[2] · Santanu Kumar Rath[1] · Ramesh Dharavath[1]

Received: 14 February 2019 / Accepted: 29 August 2019 / Published online: 6 September 2019


**Abstract**
Communities in social networks are the essential feature which may be considered as a potential parameter in modeling the behavior of the social entities. Detection of communities has attracted a lot of attention in research in social network analysis. It is one of the major challenging problems as it involves high complexity in processing complex web structure. In fact, this problem can be considered as a NP-complete problem in large-scale networks, as this problem is somewhat reducible to the clique problem in graph theory. A number of meta-heuristic algorithms have been proposed to explore the hidden communities. Most of these algorithms have considered the modularity of the network as their objective function. But, the aspect of optimizing the value of modularity is associated with a problem known as resolution limit, where the size of the detected communities depends on the number of edges existing in the network. In this paper, a genetic algorithm-based community detection has been proposed where an efficient single objective function based on similarity matrix has been devised. The similarity index between each pair of nodes has been calculated in a distributed manner over multiple computing nodes. Similarity index proposed in this paper is based on the topological structure of the network. The effectiveness of the proposed approach is examined by comparing the performance with other state-of-the-art community detection algorithms applied over some real-world network datasets.

**Keywords** Genetic algorithm · Community detection · Similarity index · Fitness function

## 1 Introduction

In the real world, the entity of a complex structure can be represented as a network structure. Communities are the basic building blocks in network evolution. They usually resemble with functional units in the complex systems. By analyzing the communities, one can get the detailed insight about the functionality of large-scale system. However, detecting communities in the large-scale network is a challenging task due to its heterogeneous nature and high complexity. It is also due to the existence of several definitions for community based on different features and characteristics. As per the terminology, communities are the group of nodes that are densely connected within the group [1]. Connections among the nodes indicate the similarity between the nodes. Similarities between the nodes are based on either topological features or content features. Similarities between the nodes within the community tend to be more similar as compared to nodes outside the community.

Traditional community detection algorithms fail to identify all the hidden communities in large-scale networks in reasonable amount of time. In fact, the problem is found to be a NP-complete [2]. Research on community detection has received a lot of attention in various application domains. For example, communities in biological network may correspond to protein complex unit. In social network,

✉ Ramesh Dharavath
drramesh@iitism.ac.in

Ranjan Kumar Behera
jranjanb.19@gmail.com

Debadatta Naik
deba.uce03@gmail.com

Santanu Kumar Rath
skrath@nitrkl.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, India

[2] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad 769008, India

it may be a group formed by the users based on common interest. Communities in the network are the higher-order substructure within the network. Identifying and analyzing these substructures may help in understanding the inherent features and predicting the behavior of the complex systems. The approach to solve the community detection problem can either be heuristic based or optimization based. Heuristic-based community detection algorithms are often based on few assumptions such as in Girvan Newman algorithm, where the assumption is that the edges in inter-communities have higher edge-betweenness value as compared to edges in intra-communities [3]. Communities can be identified sequentially by removing the edges in non-increasing order of the edge-betweenness value. Optimization-based community detection algorithms deal with maximizing the objective function that captures the structural information in the network [4]. However, the algorithms based on the maximizing modularity expose with two major problems. First, computing modularity value for all the possible combination of partition is a NP-hard problem, and second, modularity-based algorithm often is prone to resolution limits, which proves that the detected community size depends on the size of the network, and few small-sized communities may not be identified in the detection process.

In order to handle the NP-completeness of the problems, genetic algorithm has been considered as one of the suitable approaches to obtain an optimal result. The methodology of GA is inspired by the biological evolution of genes to obtain better offspring. Standard genetic algorithms randomly initialize set of candidate solutions for the problem, where an objective function is defined to measure the fitness value for each solution. Set of solutions having better fitness value are considered for processing the next generation sequences. Also, a set of new offsprings are generated based on the genetic operator. To solve the related issues, in this paper, genetic algorithm has been applied to identify and analyze the community structure in large networks. Depending on the structure of the network, it is often partitioned into different number of communities. In this study, a new objective function has been proposed to measure the fitness value for each chromosome.

The subsequent section of the paper is presented as follows: Sect. 2 presents the related literature about the community detection problem. Section 3 briefly explains the problem statement of the work. Similarity index has been considered as the major parameter in evaluating fitness quality. Different similarity indices are presented in Sect. 4. The proposed community detection approach based on genetic algorithm is discussed in Sect. 5. Section 6 presents the quality score to evaluate the effectiveness of the algorithm. Experimental setup, detail of the dataset used for implementation and algorithms used for comparison are presented in Sect. 7. Section 8 presents the limitations of the proposed approach. Finally, conclusion and future work are discussed in Sect. 9.

## 2 Related literature

Several algorithms have been proposed to identify the communities in large-scale networks as reported in the literature [5–8]. The most commonly accepted one was proposed by Newman and Girvan, which is similar to the approach followed on the hierarchal clustering in data mining [9]. In this algorithm, communities were identified after the removal of edges in the non-increasing order of their edge-betweenness value. The edge-betweenness measure is defined as the ratio of number of shortest paths (between any two nodes), that are passed through it, to the total number of shortest paths in the network. The assumption taken in Newman and Girvan algorithm is that the edges, present between the communities, are having higher edge-betweenness value [10]. On the other hand, in case of weighted graph, the edge betweenness value depends on the content associated with the links [11]. By identifying and removing these edges, one can identify the hidden communities. However, in their algorithm edge-betweenness value is recalculated after the removal of each edge which is a very time-consuming process, especially for large-scale networks. In this paper, we have tried to reduce the complexity of the problem by applying meta-heuristic approaches such as genetic algorithm. Clauset et al. [12] proposed a community detection algorithm which is based on the greedy paradigm. It is based on the agglomerative approach, where initially each node in the network is considered as a community. Communities are then merged on the basis of a measure known as modularity (Q). It is defined as the difference between inter-community edge-density and intra-community edge-density. It depends on two structural parameters $e_{ij}$ and $a_i$ which are defined as follows:

$$e_{ij} = \frac{1}{2m}\sum_{vw} A_{vw} \tag{1}$$

and

$$a_i = \frac{1}{2m}\sum_{vw} k_{vw} \tag{2}$$

Here, $e_{ij}$ measures the fraction of inter-community edges and $a_i$ measures the intra-community edges in the network. The modularity proposed by the author Clauset et al. [12] can be expressed as:

$$Q = \frac{1}{2m}\sum\nolimits_{vw}\left[A_{vw} - \frac{k_v k_w}{2m}\right]\delta(C_v, C_w) \tag{3}$$

where $m$ is the total number of edges in the network $A_{vw}$ is 1, if an edge exist between node $v$ and node $w$. Otherwise, its value is 0. $\frac{k_v k_w}{2m}$ is the probability of having an edge between node $v$ and node $w$ with degree $k_v$ and $k_w$, respectively, if edges are randomly assigned in the network. $\delta(C_v, C_w)$ is one, if both $v$ and $w$ belong to the same community and it is zero, if they belong to the different communities. The performance of the algorithm depends on the efficient computation of the modularity. It also suffers from a problem known as resolution limit, where small-sized communities remain unidentified in the process. In this paper, we address this issue by proposing a new quality metric to quantify the community structure in the network. We name it as group similarity density. Pizzuti et al. [13] proposed a genetic algorithm for community detection in social networks. The algorithm optimizes fitness function known as community score to effectively identify the densely connected nodes. In their algorithm, each chromosome consists of vector of $n$ number of genes, where $n$ is the number of nodes in the dataset. The value assigned at each index is in the range of 1 to $n$. Here, the maximum number of solutions could be $n^n$ which is exponential in nature. Unlike the encoding scheme adopted by Pizzuti et al. [13], we have encoded our solution as vector of $n$ elements, in which the value assigned to it is in the range of 1 to $k$, where $k$ is the predefined number of communities. In our encoding scheme, the maximum number of solutions could be $k^n$. This reduces the solution space drastically, which leads to faster convergence of the algorithm. In the algorithm proposed by Clara Pizzuti et al., the number of communities detected is same as the number of connected components existing in the network. However, this could be unrealistic in real-world network as in a connected component there could be more than one community. Another algorithm proposed by Radicchi is based on two quantitative definitions [14]. A subgraph in a network is said to be a community, if it has more inter-connections as compared to the number of connections outside the subgraph in the stronger sense. On the other hand, if the sum of all in-degrees in vertex set $V$ is greater than the sum of the out-degrees, then the subgraph is said to be a community in a weaker sense. This algorithm works in a manner similar to the one proposed by Newman [9]. However, the edge with smallest value of edge-clustering coefficient is first removed. The edge-clustering coefficient of an edge $e_{uv}$ is defined as the ratio between the number of common neighbors of node $u$ and $v$ to the minimum degree of node $u$ and $v$ [15].

To overcome the existing mutation techniques used to detect the community structures in the networks, a genetic algorithm based on local search was proposed by Jin et al. [16]. In this work, local search strategy based on marginal gene concept was combined with an efficient mutation method to analyze the modularity function. Another collaborative evolutionary model was proposed by Chira and Gog to detect the communities in a network [17]. In this work, search space was guided by best–worst recombination and model was dependent on collaborative selection. The advantage of this algorithm is that it does not require prior knowledge of number of communities present in the network. On the other hand, this algorithm does not compute the overlapping community structures of the network. Gong et al. [18] proposed a single objective memetic algorithm for community detection, which optimizes the modularity density function to obtain the optimal solutions. In this approach, hill climbing strategy was used for local search procedure. In the subsequent year, Gong et al. [19] proposed an improved version of memetic algorithm by presenting population generation by means of label propagation strategy, an elitism strategy and an ISACLS local search strategy. A differential evolution-based optimization algorithm was proposed by Jia et al. [20] to detect the communities in a given network, considering modularity as fitness function. Binary crossover was used for the transmission of information in the evolution. No prior knowledge regarding the number of communities was required, and quality was increased by implementing biased initialization process and clean-up operations. Shang et al. [21] proposed the improved genetic algorithm for community detection, in which modularity function is used as the fitness function and simulated annealing procedure as local search strategy. This algorithm needs prior knowledge of number of community structures. A local search-based genetic algorithm was proposed by Liu et al. [22]. In this, concepts of marginal gene and monotonicity were used to overcome the problems associated with state-of-the-art mutation methods. Along with these concepts, local search strategy was combined to form a new mutation method. Similarly, Shi et al. [23] developed a genetic algorithm-based strategy that uses genetic operations to cluster the links associated among nodes of the network. Since a node can be a member of more than one cluster, the proposed algorithm was able to identify the overlapping community structures. A knowledge-based evolutionary algorithm was proposed by Zadeh et al. [24] to detect the communities in a network. Extracted knowledge from network was used to obtain the optimal solutions and search strategy. Knowledge has been updated in each progression depending on the present status of the network. A glance of comparative analysis between the single objective methods and our algorithm is present in Table 1.

Gupta et al. [25] proposed a parallel execution of community detection algorithm, which uses variations of the outstanding quantum-inspired evolutionary algorithm (QIEA). QIEA algorithm is likewise described by the

**Table 1** Comparison of single objective function

| Authors | Year | Fitness function | Representation | Mutation | Crossover | OC | Distributed computing |
|---|---|---|---|---|---|---|---|
| Jin et al. [16] | 2010 | Modularity (Q) | Locus based | Neighbor | Uniform | N | N |
| Chira and Gog [17] | 2011 | Community score (CS) | Locus based | Random | Collaborative | N | N |
| Gong et al. [18] | 2011 | Modularity density (D) | Label based | Neighbor | Two-way | N | N |
| Gong et al. [19] | 2012 | Modularity density (D) | Label based | Neighbor label | Two-way | N | N |
| Jia et al. [20] | 2012 | Modularity (Q) | Label based | Rand/1 | Binary | N | N |
| Shang et al. [21] | 2013 | Modularity (Q) | Label based | Random | Two-way | N | N |
| Liu et al. [22] | 2013 | Modularity (Q) | Locus based | Neighbor | Uniform | N | N |
| Shi et al. [23] | 2013 | Modularity (Q) | Locus based | Neighbor | Uniform | Y | N |
| Zadeh et al. [24] | 2015 | Community score (CS) | Locus based | Neighbor | Uniform | N | N |
| Proposed method (GA-BCD) | 2019 | Group similarity density (Z) | Label based | Random | Multi-point | N | Y |

OC, overlapping community; Q, modularity; CS, community score; D, modularity density; Z, group similarity density

A set of communities are said to be overlapping communities (OC) if they are sharing at least one common node. The communities $C_i$ and $C_j$ are said to be overlapping with each other if $C_i \cap C_j \neq \emptyset$, where $i \neq j$

Community score (CS) is one of the quality measures used to quantify the community partition adopted by many authors in the literature. It is defined as the sum of scores corresponding to each of the communities explored in the community partition. They have measured the score by maximizing the in-degree of individual community. Modularity density (D) is the modified version of the standard modularity, which was defined by Clauset et al. [12]. The prime objective of modularity density was to overcome the resolution limit problem existing in standard modularity. We have proposed a different quality measure to quantify the community partition obtained through genetic algorithm. We name it as group similarity density (Z). The details of this measure are discussed in Sect. 3

individual's representation, population dynamics and the evaluating function. The algorithms are parallelized at thread as well as block level. It is able to achieve the maximum modularity value for the identified community structures, and computation is faster than serial version of algorithms. Zhang et al. [26] proposed a faster and mixed representation evolutionary algorithm which is able to extract the overlapping communities in the complex networks. Mixed representation algorithms consist of two categories of nodes: one the nodes which are common in more than one community and others are the nodes which are found in a single community. Updating individual methodologies are applied in two categories of nodes. The performance of the proposed algorithm is effective and compared with six other overlapping community detection algorithms. Guerrero et al. [27] proposed a new general genetic algorithm (GGA+), which adopts a flexible and adaptive analysis of the network structure. Under the guidance of modularity, efficient strategies for initialization and search operators are included in GGA+ algorithm to evaluate the network structure. This approach is efficient and scalable in nature. Wen et al. [28] proposed a maximal clique-based evolutionary algorithm for community detection. They have considered a multi-objective function for the detection of overlapping communities. Maximal clique graph provides the intrinsic property called overlapping of nodes. Although maximal clique represents community in a strong sense, but the nodes may not be mapped to the actual community. As a result, quality of the community partition may be degraded. Guendouz et al. [29] attempted to detect the communities with best solutions without considering the structure of the system. Here, author used the fireworks algorithm (FWA) that consists of new initialization and mutation methodologies. This property increases the algorithm's speed of convergence. Chen et al. [30] adopted a novel similarity measure to detect the communities for signed network. The primary procedure of the algorithm depends on the reconstructed neighbor sets, which are found by the similarity measure. The continuous change in states of nodes is imitated by differential equation. The performance of the proposed algorithm is high. Ju et al. [31] proposed a membrane-based multi-objective evolutionary algorithm to detect communities in networks, where the whole population is separated into different membranes. Here, the value of two evaluating parameters named Ratio Cut and Kernel J-means are minimized. The algorithm is time efficient and diverse in nature. Some of the other meta-heuristic algorithms such as ant colony, bee colony and bat algorithm are also applied in several papers for community detection algorithms. Rani et al. [32] have proposed a hybrid version of bat algorithm which employs Tabu search strategy for obtaining better solution for discrete optimization problems such as community detection. In this work, the authors have shown that Tabu search strategy is the promising approach to enhance the ability of local search for bat

algorithm. Behera et al. [33] have presented a parallel version of community detection algorithm for small world network. In this work, the authors have considered the concept of six degrees of separation to explore the communities in a real-world complex network. Moreover, the algorithm has been implemented in Hadoop distributed platform with MapReduce framework. Ji et al. [34] have considered the social communities based on user rating and product reviews for improving recommendation accuracy. They have proposed the hybrid recommendation system by including social communities identified from the social media reviews. In this paper, non-overlapping and overlapping social communities have been detected using CNM and CoDA community detection algorithms, respectively. An extensive survey of different community detection algorithms has been presented by authors Azaouzi et al. [35]. In their paper, the authors classified all the community detection algorithms based on either distributed or centralized computation. They have also presented the algorithms that are suitable for both static and dynamic networks.

## 3 Problem statement

In this section, we represent the network structure as follows. Complex network can be represented in the form of a graph $G = (V, E)$, where $V$ represents set of entities in the network and E represents set of relationships between the entities and $n = |V|$ and $m = |E|$. Nodes within a community tend to be more similar as compared to nodes outside the community.

In community detection problem, the objective is to find out $k$ subgraphs or a cover of $k$ communities which are represented as follows:

$$C = \{C_1, C_2, C_3 \ldots, C_k\} \quad \text{and} \quad |C| = k \tag{4}$$

and

$$V = \bigcup_{i=1}^{k} V_i \quad \text{and} \quad E = \bigcup_{i=1}^{k} E_i + e \tag{5}$$

where $V_i$ and $E_i$ are the set of vertices and edges in the $i$th community and $e$ is the set of inter-community edges. The detected cover of $C$ is said to have disjoint communities, if $C_i \cap C_j = \emptyset, \forall (i,j) \in [1,k]$, otherwise it is said to have overlapping communities.

In this study, we represent community partition in the form of membership vector which is a numeric vector. Each index in the vector represents an entity in the network, and each element represents its community membership in the partitioned network. The objective is to find out the membership vector of the network that represents maximum group similarity density of the network. It can be modeled as:

$$\arg\max Z = \left(1 + \frac{1}{\sqrt{k}}\right) \prod_{i=1}^{k} \mathrm{SD}_i \tag{6}$$

Here, the factor $\left(1 + \frac{1}{\sqrt{k}}\right)$ is considered as the adjustment factor in order to maintain the divergence of nodes into more than the expected community. Higher the number of communities, more will be the $Z$ value. $\mathrm{SD}_i$ is the similarity density for the $i$th community. Similarity, density for each community can be obtained by adding the similarity value for all pairs of nodes within the community. It can be obtained by the following equation:

$$\mathrm{SD}_i = \sum\nolimits_{a,b \in C_i} S(a,b) \tag{7}$$

where $a$ and $b$ are the entities in the $i$th community and $S(a, b)$ is the similarity value between $a$ and $b$. Problem of community detection can be reduced to an optimization problem that seeks to maximize the value of $Z$ in Eq. 6.

## 4 Similarity index

Users in each social community tend to have certain interest in common. Similarity between two entities in a network can be measured based on various parameters. These parameters depend on either topological structure of the network or attributes associated with the nodes. In this study, similarity between each pair of nodes has been evaluated based on network structure around the nodes. All the similarity measures considered in this paper are based on the acquaintance model in sociology [36]. The intuition behind this model is that more the number of common friends, higher the chances for them to be familiar with each other. Five different similarity measures have been considered for evaluating the degree of similarity between the entities. These are defined as follows:

### 4.1 Jaccard index

Paul Jaccard proposed the statistic instance for measuring the similarity between the set of sample datasets [37]. Jaccard similarity between nodes $a$ and $b$ is defined as the ratio of common neighbors of $a$ and $b$ to the total neighbors of $a$ and $b$. It can be defined as:

$$\mathrm{Jaccard}(a,b) = \frac{|\mathrm{Nb}(a) \cap \mathrm{Nb}(b)|}{|\mathrm{Nb}(a) \cup \mathrm{Nb}(b)|} \tag{8}$$

where $\mathrm{Nb}(a)$ and $\mathrm{Nb}(b)$ represent the set of neighboring nodes of $a$ and $b$, respectively. The value of Jaccard similarity is normalized between 0 and 1.

### 4.2 Simpson index

Simpson similarity index is defined as the ratio of common neighbor to the minimum degree. Unlike Jaccard similarity index, Simpson index is normalized to the minimum degree of nodes. It is similar to the topological overlap coefficient as pointed out by [38]. It is defined as:

$$\text{Simpson}(a,b) = \frac{|\text{Nb}(a) \cap \text{Nb}(b)|}{\min\{|\text{Nb}(a)|, |\text{Nb}(b)|\}} \tag{9}$$

### 4.3 Geometric index

Geometric similarity index can be calculated by dividing the square of number of common neighbors to the product of degree of both the nodes [37]. It is expressed as:

$$\text{Geometric}(a,b) = \frac{|\text{Nb}(a) \cap \text{Nb}(b)|^2}{|\text{Nb}(a)| \times |\text{Nb}(b)|} \tag{10}$$

### 4.4 Cosine index

Salton proposed the cosine similarity index which has been widely used in a number of applications [39]. Mostly, they are helpful in finding the similarity between citation networks. It is defined as the cosine angle between two sample vectors. It can be expressed as the ratio of number of common neighbors to the square root of the product of degree. In this paper, vector of neighboring nodes is considered as the sample vector. It is presented as:

$$\text{Cosine}(a,b) = \frac{|\text{Nb}(a) \cap \text{Nb}(b)|}{\sqrt{|\text{Nb}(a)| \times |\text{Nb}(b)|}} \tag{11}$$

### 4.5 Sorenson index

Sorenson proposed a similarity measure similar to Jaccard index to measure the similarity between the species [37]. It has been widely used in social network analysis. It is defined as the ratio of twice the number of common neighbor to the sum of degree of the nodes. It is expressed as follows:

$$\text{Sorenson}(a,b) = \frac{2 \times |\text{Nb}(a) \cap \text{Nb}(b)|}{|\text{Nb}(a)| + |\text{Nb}(b)|} \tag{12}$$

## 5 Proposed methodology: GA-based community detection

The performance of genetic algorithm mainly depends on the operators and the fitness function associated with the problem. The proposed genetic algorithm-based community detection (GA-BCD) has been presented in Algorithm$_1$.

**Algorithm$_1$: GA-BCD Algorithm for Community Detection**

**Input**: The network G = (V, E) where n=|V| and e=|E|, mutation probability (m), crossover probability(p), No. of generation.

**Output**: Set of numeric vectors of size n, each represents a community partition of the network.

**Step$_1$:** Randomly generate the population by considering the safe and feasible initialization.

Safe initialization: same community not to be allocated to two nodes if they are not connected in original network.

**Step$_2$:** Calculate the similarity density of each community in a given chromosome as discussed in equation 7.

**Step$_3$:** Calculate the fitness value of each chromosome by using the following objective function.

$$\arg\max Z = \left(1 + \frac{1}{\sqrt{k}}\right) \prod_{i=1}^{k} SD_i$$

**Step$_4$:** Apply the μ+λ selection strategy in order to eliminate the least optimal solution.

**Step$_5$:** Crossover and mutation operators are applied to λ number of chromosomes in order to generate the population for next generation.

**Step$_6$:** Repeat the Step 2 to Step 5 for 100 number of generation.

**Step$_7$:** Return the community structure of the network having highest fitness value.

### 5.1 Genetic representation

Encoding scheme for genetic algorithm plays a vital role in finding the efficiency of the algorithm. In this paper, numerical encoding scheme is used for each individual chromosome. Each chromosome in the population is represented as a vector of $n$ elements, where $n$ is the number of users given in the network. Each position $i$ in the vector corresponds to a user in the network. The value at the $i$th index of the vector represents the identity of the community, where the $i$th user belongs to. Each gene can assume a value in the range 1 to $k$, where $k$ is the number of communities in the network. In this paper, community analysis for the network has been performed by predefining the number of communities in the network. Figure 1 presents the representation for a candidate solution. In this figure, $C_1$, $C_2$, $C_3$ are the three detected communities. So the range of the allele for candidate solution is 1–3.

### 5.2 Initial population generation

The convergence of genetic algorithm is often sensitive to the quality of initial population. More diversity and high average fitness value of the population converge to better partition of communities. In most of the real-world networks, the number of communities to be formed is known. In this experiment, we have predefined the number of communities in order to reduce the search space. Chromosomes are generated randomly at each generation. At the same time, in-feasible solutions are prevented by safe initialization where two nodes are kept in one community only if they are connected. The disconnected nodes are checked at the initialization process of individual chromosomes. The value for each gene is in the range of 1 to $k$, where $k$ is the number of predefined communities. The performance of the algorithm is evaluated for different values of $k$ for each of the datasets. The $k$ value has been considered in the range of 2–10. For example, as shown in Fig. 1, a candidate solution for a network having ten number of nodes is presented. At the same time, number of communities to be partitioned, i.e., $k$, is considered to be 3. The value present in the candidate solution is in the range of 1–3. Similar to the example shown in Fig. 1, the chromosome sets are generated for each value of $k$. The length of the candidate solution depends on the number of nodes in each dataset. For each of the $k$ value, initially 100 number of candidate solutions have been generated randomly.

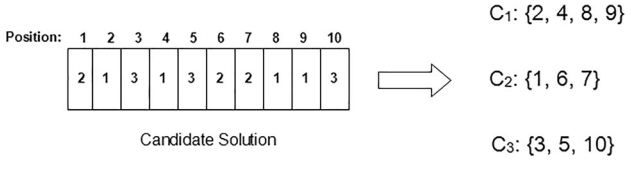


**Fig. 1** Genetic representation for candidate solution

### 5.3 Proposed fitness function

To validate the proposed algorithm, population size has been considered as 100, i.e., one hundred number of chromosomes have been randomly generated at the first generation. As they are randomly generated, most of them may not be able to provide feasible solution for community detection problem. Therefore, suitable candidate solutions are to be chosen based on the fitness value which has been determined by the fitness function. The proposed fitness function is defined as:

$$\arg\max Z = \left(1 + \frac{1}{\sqrt{k}}\right) \prod_{i=1}^{k} \mathrm{SD}_i \tag{13}$$

where $\mathrm{SD}_i$ is the similarity density for the $i$th community, i.e., $C_i$. It is described in Eq. 7.

Fitness value of individual solution seems to be increased with the number of generations. Figure 2 represents the change in fitness value with the generation number for five different datasets. Fitness value changes rapidly after 40–50 generations for most of the dataset. In this study, experiment has been performed until the graph is partitioned into ten numbers of communities. It is observed from Fig. 2 that change in fitness value is more prominent if the graph is to be partitioned into more number of communities. On the other hand, change in fitness value is minimal, when the graph is partitioned into less number of communities. The parameters used in GA are shown in Table 2.

### 5.4 Selection

Selection is one of the crucial phases in genetic algorithm which helps in global search for the best solution of the problem. Before the selection process starts, each individual is assigned a rank according to their fitness value. The fitness value for each individual is calculated based on the objective function as defined in this problem. In this work, $\mu + \lambda$ selection strategy has been adopted for selecting chromosomes for further generation. $\mu + \lambda$ selection strategy is one of the preferred methods for solving evolutionary algorithms. As per this strategy, $\mu$ number of best chromosomes from the current population is to be kept for further processing. $\lambda$ is the number of offsprings which are generated from the fittest parents through genetic operators such as crossover and mutation. At the same time, $\lambda + \mu$ number of chromosomes are considered for solution in the next generation.

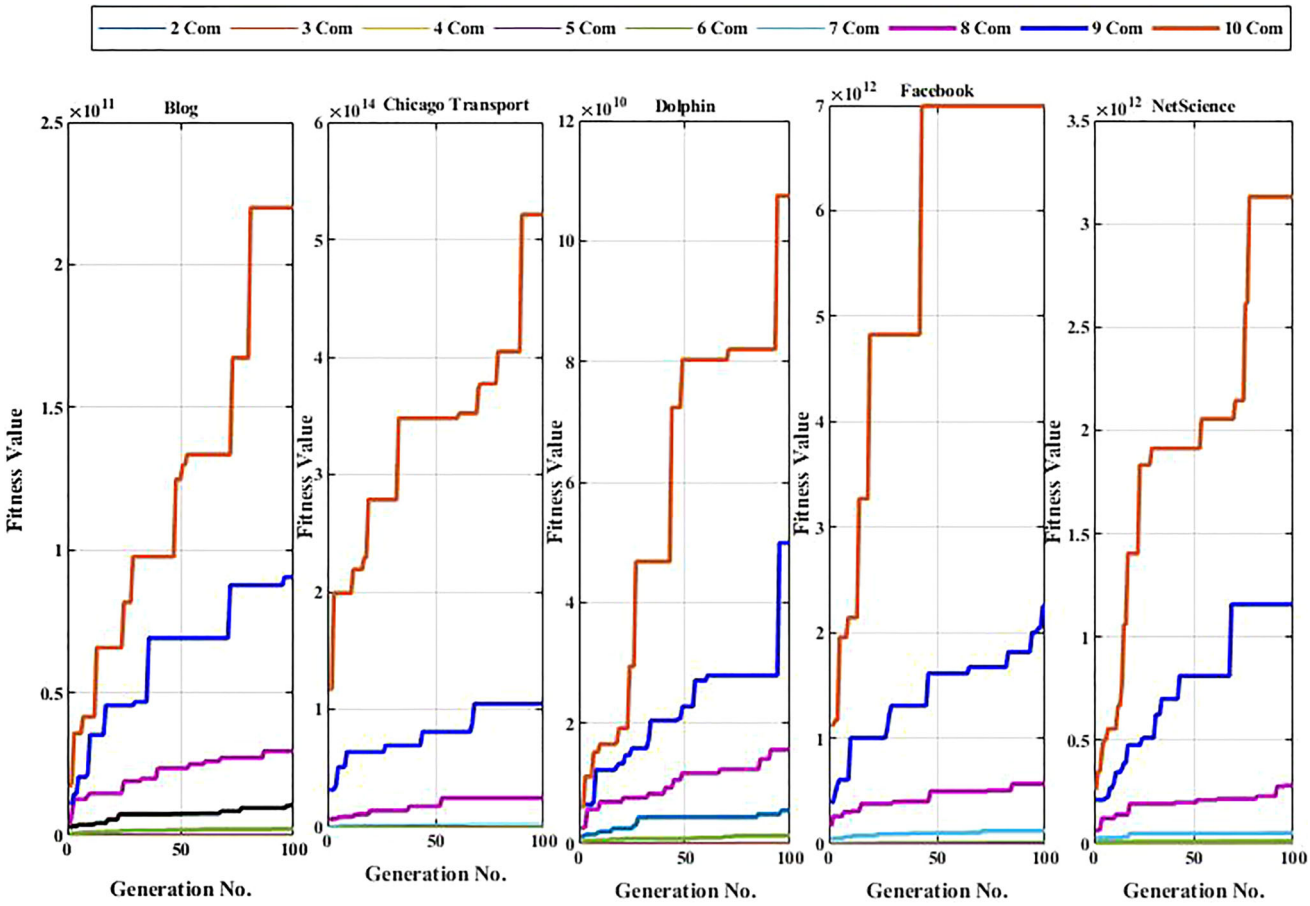


**Fig. 2** Change in fitness value in different number of generations

**Table 2** Parameters used in genetic algorithm

| Parameter | Value | Description |
|---|---|---|
| $C_{\text{length}}$ | No. of nodes in the dataset | Chromosome length |
| $P_{\text{size}}$ | 100 | Population size |
| $P_{\text{m}}$ | 0.3 | Mutation probability |
| $P_{\text{c}}$ | 0.6 | Crossover probability |
| gen | 100 | Number of generation |

## 5.5 Crossover

Crossover is the genetic operator used to perform operation on two chromosomes at a time. It combines the features of both the chromosomes to create a new chromosome for obtaining an optimal solution. The number of chromosomes that undergoes with crossover operation is determined by a parameter known as crossover probability. This operation is associated with a parameter known as crossover probability. In our experiment, crossover probability is chosen as 0.6 which means 60% of each population is to be participated for crossover at each generation.

In the proposed work, each community has been identified by a numerical identifier. Each chromosome is an array of numerical values which represent the community membership or identifier for individual entity. The index of the array represents the node identity in the network. The same community identifier in different chromosomes may correspond to different communities, or different community identifiers in different chromosomes may correspond to the same community. For example, (5,5,5,1,1,1) and (1,1,1,5,5,5) represent the same network partition, but the community whose identity is 5 in the first chromosome corresponds to the same community with identifier 1 in the second chromosome. If the crossover point happens to be at 1, the resulting chromosome will be (5,5,5,5,5,5) and (1,1,1,1,1,1). Still both the chromosomes will have the same community structure. To avoid the above occurrences, multi-point crossover has been adopted for generating chromosomes. In multi-point crossover, community identities will be exchanged between the chromosomes at multiple points. Position of the crossover is chosen randomly at each iteration.

### 5.6 Mutation

Genetic algorithm contributes a powerful search capability. Mutation operator attains its local search function by shifting a node from one community to another community, and crossover operator attains its global search function by combining and splitting the communities. In this work, multi-point crossover strategy achieves its objective. The traditional mutation operators fail to achieve the target function by shifting a node between the communities, because these operators may combine or spilt the communities, which is strongly undesirable.

In the proposed approach, $\arg\max Z = \left(1+\frac{1}{\sqrt{k}}\right)\prod_{i=1}^{k} \mathrm{SD}_i$ is the objective function and $\mathrm{SD}_i = \sum_{a,b\in C_i} S(a,b)$ is the sum of similarity measure of each pair of nodes of a community. This similarity density function plays the role of local function from the local point of perspective of the node. The value of $Z$ increases with the local function $\mathrm{SD}_i$ without changing the label of the other nodes. By this way, the local search strategy works and shifts a node between the communities which resulted in better solutions.

## 6 Proposed quality score

Community has been detected based on each similarity matrix for five real-world datasets. At each generation, the proposed genetic algorithm optimizes the set of 100 community partitions based on fitness value. Communities obtained at the 100th generation are considered to be possible list of optimal candidate solutions. Individual with the highest fitness value at the 100th generation is selected as the final partition of the network. In this study, an efficient measure known as quality score has been proposed to evaluate the final communities obtained through different similarity indices. The intuition behind quality score is to find out the ratio of inter-community dissimilarity to the intra-community dissimilarity. It is expressed as:

$$\text{Quality Score (QS)} = \frac{S'_{\text{inter}}}{S'_{\text{intra}}} \tag{14}$$

where $S'_{\text{inter}}$ and $S'_{\text{intra}}$ are dissimilarity indices between the nodes in different communities and nodes in the same communities, respectively. $S'_{\text{inter}}$ and $S'_{\text{intra}}$ are mathematically expressed as follows:

$$S'_{\text{inter}} = \sum_{a\in C_i, b\in C_j} \frac{1}{S_{ab}},\ C_i, C_j \in C \quad \text{and} \quad i \neq j \tag{15}$$

and

$$S'_{\text{intra}} = \sum_{a,b\in C_i} \frac{1}{S_{ab}}, \quad C_i \in C \tag{16}$$

Here, $a$ and $b$ are the nodes in the network and $C$ is the set of communities identified through the proposed genetic algorithm. The objective is to maximize the quality score for better partition of the network. Higher the quality score, better the quality of detected community. Network partition with highest quality score is considered to be an optimal solution for the community detection problem (Table 3).

## 7 Experimental analysis

Heuristic-based algorithms are often suitable for the complex problems, where computation is highly expensive than expectation. The proposed GA-BCD algorithm has been performed in the distributed platform on five real-world network datasets. The details of dataset description, experimental setup and analysis are provided in following subsections.

### 7.1 Dataset description

In this paper, the following datasets are used for measuring the performance metrics. Details of these datasets are listed in Table 4. Network dataset can be depicted as graph structure, where each node represents an entity and links between the nodes represent the relationships between the entities. Each of the following datasets is collected in the form of edge list.

1. Blogs [40]: This dataset is collected from the context of 2004 US-election, where nodes represent the blogs and edge represents the hyperlinks between them. One blog may have page with hyperlink that points to other blogs, while reverse link may not exist in the dataset.
2. Chicago transport [41]: In this dataset, road transportation of Chicago is represented as network structure, where node represents the place in Chicago and edge represents the connections between the places.
3. Dolphin [42]: This dataset is created by observing few groups of dolphin communities during 1994 to 2001 living in New Zealand, where dolphins and frequent interactions between dolphins are represented as nodes and edges, respectively. It is a kind of social network having directed edges.
4. Facebook [43]: Facebook is the most popular social network platform for social interaction. A small part of this network has been collected from website https://snap.stanford.edu/. The nodes and edges represent the social user and their relationship, respectively.

**Table 3** Quality score of community partition of different datasets for different similarity values

| SI | No. of comm. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| *Blogs* | | | | | | | | | |
| Jaccard | 0.316 | 0.525 | 0.665 | 0.796 | 0.860 | 0.914 | 0.989 | 0.998 | 1.051 |
| Simpson | 0.516 | 0.860 | 1.205 | 1.443 | 1.642 | 1.777 | 1.873 | 2.005 | 2.125 |
| Geometric | 0.273 | 0.454 | 0.602 | 0.674 | 0.766 | 0.811 | 0.863 | 0.879 | 0.908 |
| Cosine | 0.420 | 0.690 | 0.940 | 1.085 | 1.203 | 1.304 | 1.364 | 1.442 | 1.502 |
| Sorenson | 0.393 | 0.638 | 0.836 | 0.991 | 1.091 | 1.200 | 1.259 | 1.300 | 1.344 |
| *Chicago transport* | | | | | | | | | |
| Jaccard | 0.636 | 1.147 | 1.567 | 1.884 | 2.184 | 2.488 | 2.675 | 2.856 | 3.061 |
| Simpson | 0.626 | 1.145 | 1.544 | 1.903 | 2.172 | 2.465 | 2.688 | 2.845 | 3.058 |
| Geometric | 0.637 | 1.147 | 1.564 | 1.892 | 2.157 | 2.483 | 2.643 | 2.875 | 3.067 |
| Cosine | 0.631 | 1.163 | 1.548 | 1.918 | 2.201 | 2.441 | 2.660 | 2.856 | 3.018 |
| Sorenson | 0.635 | 1.152 | 1.548 | 1.894 | 2.240 | 2.472 | 2.663 | 2.912 | 3.043 |
| *Dolphin* | | | | | | | | | |
| Jaccard | 0.097 | 0.218 | 0.394 | 0.543 | 0.485 | 0.665 | 0.788 | 0.854 | 0.845 |
| Simpson | 0.175 | 0.467 | 0.762 | 0.923 | 1.064 | 1.253 | 1.401 | 1.521 | 1.479 |
| Geometric | 0.076 | 0.217 | 0.366 | 0.385 | 0.435 | 0.513 | 0.616 | 0.705 | 0.706 |
| Cosine | 0.127 | 0.386 | 0.608 | 0.732 | 1.003 | 1.067 | 1.080 | 1.227 | 1.440 |
| Sorenson | 0.132 | 0.288 | 0.560 | 0.656 | 0.897 | 0.954 | 1.110 | 1.177 | 1.116 |
| *Facebook* | | | | | | | | | |
| Jaccard | 0.002 | 0.002 | 0.010 | 0.010 | 0.011 | 0.016 | 0.019 | 0.020 | 0.018 |
| Simpson | 0.007 | 0.010 | 0.014 | 0.019 | 0.023 | 0.027 | 0.035 | 0.032 | 0.037 |
| Geometric | 0.002 | 0.002 | 0.010 | 0.010 | 0.013 | 0.015 | 0.020 | 0.020 | 0.020 |
| Cosine | 0.003 | 0.003 | 0.011 | 0.011 | 0.015 | 0.016 | 0.023 | 0.023 | 0.020 |
| Sorenson | 0.002 | 0.002 | 0.011 | 0.011 | 0.012 | 0.017 | 0.021 | 0.021 | 0.020 |
| *Netscience* | | | | | | | | | |
| Jaccard | 0.396 | 0.697 | 0.853 | 1.036 | 1.132 | 1.248 | 1.318 | 1.377 | 1.430 |
| Simpson | 0.537 | 0.970 | 1.292 | 1.602 | 1.809 | 2.008 | 2.161 | 2.286 | 2.400 |
| Geometric | 0.384 | 0.646 | 0.815 | 0.943 | 1.055 | 1.147 | 1.217 | 1.245 | 1.325 |
| Cosine | 0478 | 0.845 | 1.130 | 1.337 | 1.514 | 1.620 | 1.744 | 1.826 | 1.934 |
| Sorenson | 0.463 | 0.834 | 1.085 | 1.281 | 1.444 | 1.574 | 1.652 | 1.768 | 1.792 |

**Table 4** Datasets used

| S. no | Datasets | Node | Edges | Clustering coefficient |
|---|---|---|---|---|
| 1 | Blog [40] | 1224 | 2615 | 0.226 |
| 2 | Chicago transport [41] | 1467 | 1298 | 0.169 |
| 3 | Dolphin [42] | 62 | 159 | 0.309 |
| 4 | Facebook [43] | 4039 | 88,234 | 0.6055 |
| 5 | Netscience [44] | 1589 | 2742 | 0.427 |

5. Net science [44]: This dataset reveals the co-authorship network between the scientists who are working on network theory. The nodes correspond to scientist and the edges exist between the nodes if the corresponding scientists have jointly published a paper.

## 7.2 Experimental setup for distributed computing

Experiments are performed on a cluster of ten number of computational nodes, configured as master–slave architecture. One of them acts as master, and the rest of the computing nodes act as slaves. Each computing node is configured with i7 processor and 3.4 GHz clock speed.

They all have symmetric configuration with 1 TB hard disk and 20 GB of RAM. The job is submitted at the master node, and data are then distributed across the cluster. Each of the computing nodes performs operation on different parts of the data independently. The results from each of the computational nodes are accumulated at the master node for further analysis of the community structure. All the computational nodes process the data in a parallel manner. Complex topological structure is analyzed, and neighbors of each of the entities are revealed. Experiments are performed for partitioning the network into 2 to 10 number of communities. Similarity between all pairs of nodes has been evaluated as per the measures given in Sect. 4 for each of the datasets. Using the similarity values, fitness values of the chromosomes are calculated.

### 7.3 Comparative analysis

The proposed algorithm GA-BCD is compared with the following community detection algorithms on the five real-world datasets listed in Table 4.

***Girvan Newman*** **[5]** This is the most popular community detection algorithm, where the community is detected by optimizing the modularity value of the partition. Here, communities are identified by eliminating the edges with the highest edge-betweenness value in an iterative manner. This is similar to divisive hierarchical clustering algorithm used in data mining.

***Walk-Trap*** **[6]** The intuition behind the Walk-Trap community detection algorithm is that when random walker start visiting nodes in the network, it is more likely to get trapped in a dense region than the sparse one. Communities can be explored by analyzing the patterns of node-traversal by a random walker in the network. It is found to have better time complexity as compared to others.

***Label propagation*** **[7]** Label propagation algorithm is found to have near-linear time complexity in detecting community for large-scale network. In this algorithm, initially, each node is assigned with a unique label and the label of the node is getting updated over the processing time. The updating rule for a node is based the label of its neighboring nodes. Although it is faster, unique community partition is hard to obtain over different runs.

***Louvain*** **[8]** The Louvain community detection algorithm is similar to the Girvan Newman algorithm, where the modularity value is to be optimized over the different partitions. However, the process of the Louvain algorithm is in the reverse order of Girvan Newman algorithm. In this algorithm, initially, every node is treated as an individual community. Nodes are merged with the neighboring nodes to maximize the modularity value. This process is stopped when no further increase in modularity is possible. The whole process is similar to agglomerative clustering used in data mining.

### 7.4 Parameter evaluation

Evaluating community detection algorithms is a challenging task as the size and number of communities are often unknown in a real-world network. However, it is possible to evaluate the performance, if ground truth statistics about the community is available beforehand. The following performance parameters have been considered to evaluate the performance of GA-BCD algorithm.

***Normalized mutual information (NMI)*** It is the measure used to evaluate the similarity between two different community partitions of the same dataset. This is similar to the clustering evaluation parameter used in information theory. Danon et al. [45] used it for the first time to measure the performance of community detection algorithms. It is measured with the help of confusion matrix $C$, where the row $i$ corresponds to the true communities and the column $j$ corresponds to the found communities. Each element $C_{ij}$ corresponds to the number of nodes of community $i$ that are found in community $j$. It can be mathematically expressed as follows:

$$\mathrm{NMI}(T,F) = \frac{-2\sum_{i=1}^{|T|}\sum_{j=1}^{|F|} C_{ij}\log\left(\frac{C_{ij}C}{C_iC_j}\right)}{\sum_{i=1}^{|T|} C_i\log\left(\frac{C_i}{C}\right)+\sum_{j=1}^{|F|} C_j\log\left(\frac{C_j}{C}\right)} \tag{17}$$

where $T$ and $F$ are the actual and found community partitions, respectively. $|T|$ and $|F|$ are the number of communities present in the actual and found partitions, respectively. $C_i$ and $C_j$ are the sum of all elements in row $i$ and column $j$ of confusion matrix $C$, respectively.

***Modularity*** Modularity is the most widely used performance parameter for community detection algorithm, which quantifies the community partition obtained in a large-scale network. It is the difference between intra-community links to the inter-community links in the network partition. The mathematical definition of this parameter is described in Sect. 2.

***Adjust Rand Index (ARI)*** Rand Index is one of the important performance parameters used to measure, how much each pair of nodes is arranged same by two different clustering solutions. In other words, it is the ratio of the number of pairs of nodes belonging to same cluster or belonging to different cluster in both the partitions with to total number of pairs of nodes [46]. ARI can be mathematically represented as:

$$\mathrm{ARI} = \frac{N(P+S) - [(P+Q)(P+R) + (R+S)(Q+S)]}{N^2 - [(P+Q)(P+R) + (R+S)(Q+S)]} \tag{18}$$

where $P$ represents the number of pairs of nodes in the same cluster according to both the solutions. $Q$ represents the number of pairs of nodes in the same cluster according to solution$_1$ and these many number of pairs of nodes in different clusters according to solution$_2$. $R$ represents the number of pairs of nodes in different clusters according to solution$_1$ and these many number of pairs of nodes in same cluster according to solution$_2$. $S$ represents the number of pairs of nodes in different clusters according to both the solutions. $N$ represents the total number of pairs of nodes.

**Execution time** Execution time required to detect the communities in a large and heterogeneous network is a challenging task. It is considered to be an essential performance parameter to compare the algorithms. In this work, a comparison graph has been given for five different algorithms based on this crucial parameter.

### 7.5 Results and discussion

The performance of GA-BCD is compared with other standard community detection algorithms that are described in Sect. 7.3. Figure 3 represents the comparative analysis of different community detection algorithms based on the NMI value for the community partition. It is observed that the proposed GA-BCD algorithm performs well for large-sized datasets such as Facebook, NetScience and Blogs. The Girvan Newman algorithm was found to have better NMI in Dolphin and Chicago transport datasets. Modularity value is one of the suitable measures to quantify the community structure. Its value lies in the range between zero and one. Figure 4 represents the comparative analysis of algorithms based on the modularity value of the identified community structure. It is observed that the Walk-Trap algorithm and label propagation algorithm found to have better modularity value in Dolphin and NetScience network, respectively. ARI is another performance measure, which quantifies the similarity of node orientation in two different solutions. Comparative analysis of algorithms based on ARI value is presented in Fig. 5. It is observed that ARI value was found to be less for large-sized datasets such as Facebook. The GA-BCD algorithm has a better ARI value for Facebook and NetScience network. The Girvan Newman algorithm has better ARI value in Blogs and Dolphin network. The label propagation algorithm has shown better ARI value in the Chicago transport network. It is desirable to identify communities in a reasonable amount of time in large-scale and complex network. In this paper, a comparison based on execution time is considered to evaluate the performance of algorithms and is presented in Fig. 6. Genetic algorithm is always expected to have better execution time as compared to other traditional algorithms. From the graph, it is concluded that GA-BCD has shown better performance result in terms of execution time, especially for large datasets such as Facebook and NetScience.

The different structures of partition have been found for all the datasets and are further analyzed based on quality score which is presented in Sect. 6. Table 3 presents the quality score for community partition for different datasets after completion of the 100th epoch in genetic algorithm. Multiple set of experiments have been performed by partitioning the network into 2 to 10 number of communities. The number of communities to be partitioned is represented by each column in Table 3. Quality score quantifies the quality of community partition in the network. It is the

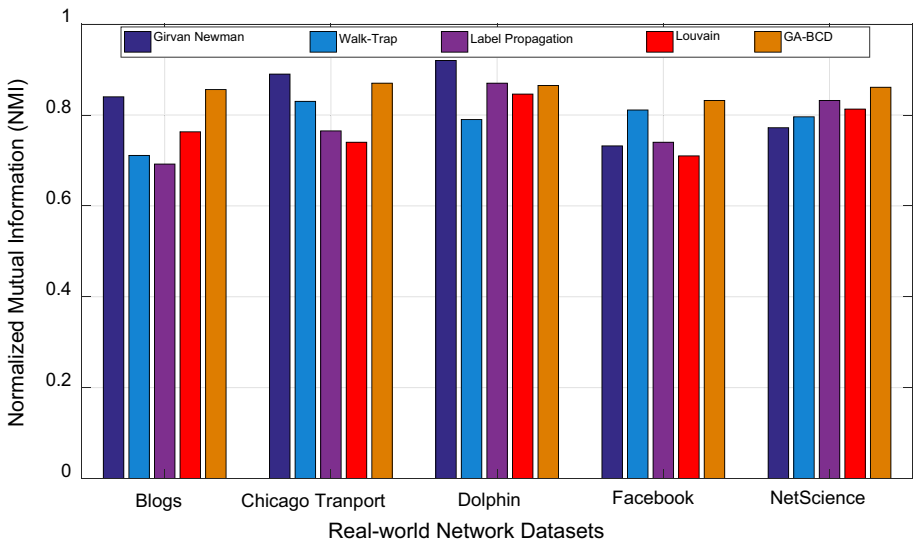


**Fig. 3** Comparative analysis based on NMI

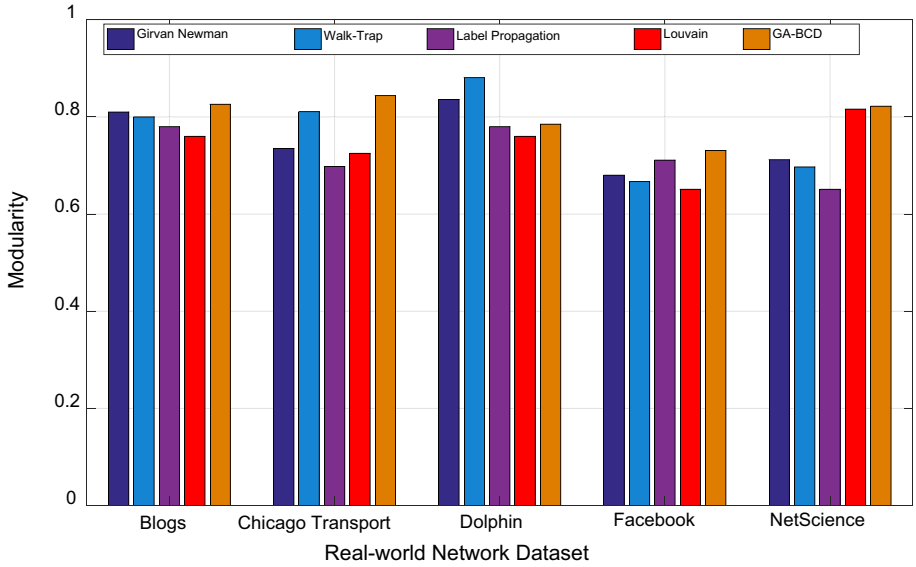


**Fig. 4** Comparative analysis based on modularity

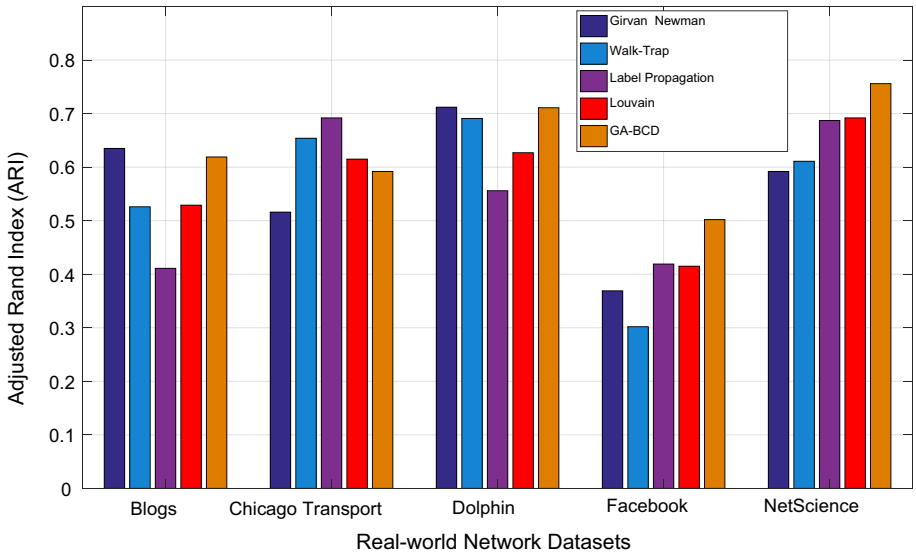


**Fig. 5** Comparative analysis based on ARI

measure of difference between intra-communities link-density and the inter-community link-density in the network. It depends on the similarity between all pairs of nodes which is based on similarity index discussed in Sect. 4. More is the quality score, better is the community partition.

It is observed from Table 3 that better partition for Blogs dataset has been obtained for Simpson similarity index and the quality score increases when the network is partitioned into more number of communities. For Chicago transport dataset, Sorenson index provides better community partition using genetic algorithm. Simpson similarity index is found to be suitable for the identification of communities in Dolphin, Facebook and Netscience datasets. Performance in terms of quality score for proposed approach is based on different number of community partitions which is presented in the form of boxplot in Fig. 7. From box plot in Fig. 7, it is observed that mean of quality score for network partition increases with the number communities. Performance based on different similarity indices is presented in the form of boxplot in Fig. 8. Although the minimum quality score is approximately equal for all similarity indices, mean quality score is better in the case of Simpson similarity index. Each similarity index has its own significance in the network structure. Their values between nodes depend on the complexity of the network. Any particular similarity index may not provide better partition in all the datasets.

Identifying community structure using different similarity values may help in deeper analysis of network structure. In this study, $P$ value and mean difference of performance have been evaluated based on the similarity

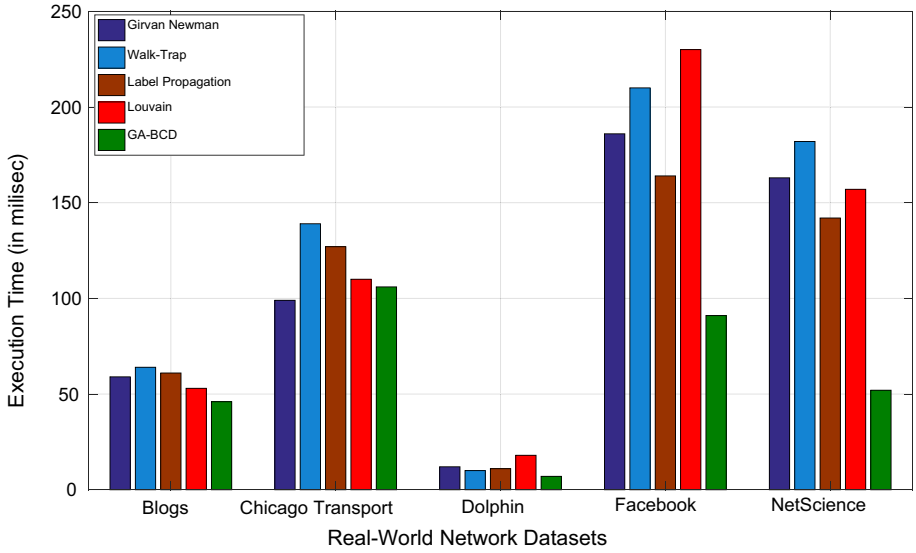


**Fig. 6** Comparative analysis based on execution time

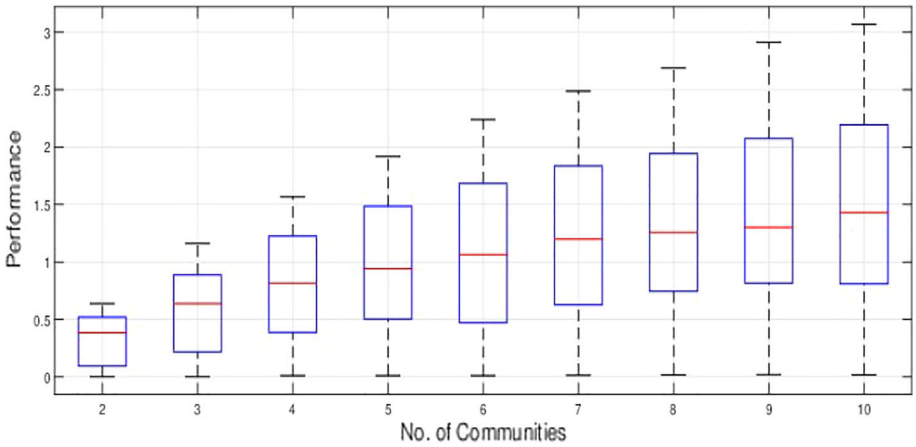


**Fig. 7** Performance of GA-BCD with different number of community partitions

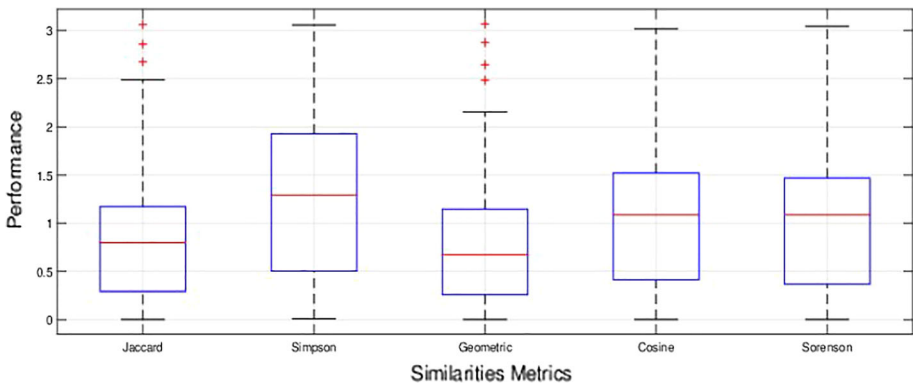


**Fig. 8** Performance of GA-BCD with different similarity indices

index. *P* value for similarity index is provided in Table 5. *P* value less than 0.001 indicates that every similarity index is significantly different from others (Table 6).

It can be concluded that application of different similarity measures for community detection is justified. *P* value and mean deviation for the result obtained in different number of communities is shown in Tables 7 and 8, respectively. From Table 7, it is observed that the model built for each of the community partitions is significantly different from each other.

**Table 5** *P* value of performance based on similarity index

| | Jaccard | Simpson | Geometric | Cosine | Sorenson |
|---|---|---|---|---|---|
| Jaccard | – | 0.00 | 0.00 | 0.00 | 0.00 |
| Simpson | 0.00 | – | 0.00 | 0.00 | 0.00 |
| Geometric | 0.00 | 0.00 | – | 0.00 | 0.00 |
| Cosine | 0.00 | 0.00 | 0.00 | – | 0.00 |
| Sorenson | 0.00 | 0.00 | 0.00 | 0.00 | – |

*P* value $\leq 0.001$ indicates the similarity indices are significantly different from each other

**Table 6** Mean Difference for similarity Index

| | Jaccard | Simpson | Geometric | Cosine | Sorenson |
|---|---|---|---|---|---|
| Jaccard | 0.000 | − 0.358 | 0.055 | − 0.189 | − 0.143 |
| Simpson | 0.358 | 0.000 | 0.413 | 0.169 | 0.215 |
| Geometric | − 0.055 | − 0.413 | 0.000 | − 0.245 | − 0.198 |
| Cosine | 0.189 | − 0.169 | 0.245 | 0.000 | 0.047 |
| Sorenson | 0.143 | − 0.215 | 0.198 | − 0.047 | 0.000 |

## 8 Threats to validity

In this work, a genetic algorithm-based community detection has been presented, where the nodes are mapped into fixed number of communities. This could be one of the limitations, since prior knowledge of number of communities is required at the encoding phase of the genetic algorithm. The proposed algorithm can be suitable for processing directed graph but may fail to process weighted network, where similarity between nodes are dependent on the strength of the relationship between the nodes. Another limitation is that it may fail to detect the overlapping communities in the network.

## 9 Conclusion and future work

Community detection is observed to be NP-complete problem, especially when processing the large-scale network. Heuristic-based algorithms are proven to be efficient means to analyze the subgroup in large-scale complex network. To accommodate this constraint, in this

**Table 7** *P* value of performance based on the number of communities

| | Com2 | Com3 | Com4 | Com5 | Com6 | Com7 | Com8 | Com9 | Com10 |
|---|---|---|---|---|---|---|---|---|---|
| Com2 | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Com3 | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Com4 | 0 | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |
| Com5 | 0 | 0 | 0 | – | 0 | 0 | 0 | 0 | 0 |
| Com6 | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 | 0 |
| Com7 | 0 | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |
| Com8 | 0 | 0 | 0 | 0 | 0 | 0 | – | 0 | 0 |
| Com9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | – | 0 |
| Com10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | – |

*P* value $\leq 0.001$ indicates that the experiments for different number of communities are significantly different from each other

**Table 8** Mean deviation for different number of communities

| | Com2 | Com3 | Com4 | Com5 | Com6 | Com7 | Com8 | Com9 | Com10 |
|---|---|---|---|---|---|---|---|---|---|
| Com2 | 0.00 | − 0.26 | − 0.48 | − 0.64 | − 0.78 | − 0.90 | − 1.00 | − 1.08 | − 1.15 |
| Com3 | 0.26 | 0.00 | − 0.22 | − 0.38 | − 0.52 | − 0.64 | − 0.73 | − 0.82 | − 0.89 |
| Com4 | 0.48 | 0.22 | 0.00 | − 0.16 | − 0.30 | − 0.42 | − 0.52 | − 0.61 | − 0.67 |
| Com5 | 0.64 | 0.38 | 0.16 | 0.00 | − 0.14 | − 0.26 | − 0.36 | − 0.44 | − 0.51 |
| Com6 | 0.78 | 0.52 | 0.30 | 0.14 | 0.00 | − 0.12 | − 0.22 | − 0.31 | − 0.37 |
| Com7 | 0.90 | 0.64 | 0.42 | 0.26 | 0.12 | 0.00 | − 0.10 | − 0.18 | − 0.25 |
| Com8 | 1.00 | 0.73 | 0.52 | 0.36 | 0.22 | 0.10 | 0.00 | − 0.09 | − 0.16 |
| Com9 | 1.08 | 0.82 | 0.61 | 0.44 | 0.31 | 0.18 | 0.09 | 0.00 | − 0.07 |
| Com10 | 1.15 | 0.89 | 0.67 | 0.51 | 0.37 | 0.25 | 0.16 | 0.07 | 0.00 |

manuscript, we have adopted genetic algorithm for community detection in real-world network. In support of this, we have proposed an objective function based on the similarity density for the community partition. The group similarity density has been measured by considering different similarity indices, which is calculated from topological information in the network. Unlike modularity, it does not suffer from resolution limit problem. We have adopted a distributed framework, where data have been processed in multiple computational nodes for faster execution. Experiments have been performed on ten different computational nodes. Networks have been partitioned into 2 to 10 number of communities. By keeping the number of communities fixed, search space of the population reduced drastically which leads to faster convergence of optimal solution. Many evolutionary algorithms for community detection available in the literature have been considered as locus-based representations, whereas we have considered label-based representation. Label-based representation provides better exploration of solution space after applying the genetic operators as compared to locus-based representation. Experimental analysis shows that GA-BCD outperforms as compared to other standard community detection algorithms in identifying the hidden communities in the network.

Community detection on real-time streaming data is found to be a challenging research problem. The work can be extended to process dynamic network, where the topological structure changes over time. Some other heuristic approaches such as particle swarm optimization and ant colony optimization can also be tested for exploring disjoint and overlapping communities.

**Acknowledgements** This research work was supported by Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions (FIST) scheme with a Grant No. SR/FST/ETI-359/2014 under Department of Science and Technology, Govt. of India. The authors wish to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India, for providing their research support.

## Compliance with ethical standards

**Conflict of interest** The authors do not have any conflict of interest.

## References

1. Nussbaum R, Esfahanian AH, Tan PN (2013) Clustering social networks using distance-preserving subgraphs. In: The influence of technology on social network analysis and mining. Springer, Vienna, pp 331–349. https://doi.org/10.1007/978-3-7091-1346-2_14
2. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174. https://doi.org/10.1016/j.physrep.2009.11.002
3. Newman ME (2004) Analysis of weighted networks. Phys Rev E 70(5):056131. https://doi.org/10.1103/PhysRevE.70.056131
4. Newman ME (2013) Spectral methods for community detection and graph partitioning. Phys Rev E 88(4):042822. https://doi.org/10.1103/PhysRevE.88.042822
5. Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826. https://doi.org/10.1073/pnas.122653799
6. Pons P, Latapy M (2005) Computing communities in large networks using random walks. In: International symposium on computer and information sciences. Springer, Berlin, pp 284–293. https://doi.org/10.1007/11569596_31
7. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106. https://doi.org/10.1103/PhysRevE.76.036106
8. De Meo P, Ferrara E, Fiumara G, Provetti A (2011) Generalized louvain method for community detection in large networks. In: 2011 11th international conference on intelligent systems design and applications (ISDA). IEEE, pp 88–93. https://doi.org/10.1109/isda.2011.6121636
9. Newman ME (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582. https://doi.org/10.1073/pnas.0601602103
10. Newman ME (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69(6):066133. https://doi.org/10.1103/PhysRevE.69.066133
11. Qi GJ, Aggarwal CC, Huang T (2012) Community detection with edge content in social media networks. In: 2012 IEEE 28th international conference on data engineering (ICDE). IEEE, pp 534–545. https://doi.org/10.1109/icde.2012.77
12. Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111. https://doi.org/10.1103/PhysRevE.70.066111
13. Pizzuti C (2008) Ga-net: a genetic algorithm for community detection in social networks. In: International conference on parallel problem solving from nature. Springer, Berlin, pp 1081–1090. https://doi.org/10.1007/978-3-540-87700-4_107
14. Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D (2004) Defining and identifying communities in networks. Proc Natl Acad Sci 101(9):2658–2663. https://doi.org/10.1073/pnas.0400054101
15. Zhang P, Wang J, Li X, Li M, Di Z, Fan Y (2008) Clustering coefficient and community structure of bipartite networks. Phys A 387(27):6869–6875. https://doi.org/10.1016/j.physa.2008.09.006
16. Jin D, He D, Liu D, Baquero C (2010) Genetic algorithm with local search for community mining in complex networks. In: 2010 22nd IEEE international conference on tools with artificial intelligence (ICTAI), vol 1, pp 105–112. IEEE. https://doi.org/10.1109/ictai.2010.23
17. Chira C, Gog A (2011) Collaborative community detection in complex networks. In: International conference on hybrid artificial intelligence systems. Springer, Berlin, pp 380–387. https://doi.org/10.1007/978-3-642-21219-2_48
18. Gong M, Fu B, Jiao L, Du H (2011) Memetic algorithm for community detection in networks. Phys Rev E 84(5):056101. https://doi.org/10.1103/PhysRevE.84.056101
19. Gong M, Cai Q, Li Y, Ma J (2012) An improved memetic algorithm for community detection in complex networks. In: 2012 IEEE Congress on evolutionary computation (CEC). IEEE, pp 1–8. https://doi.org/10.1109/cec.2012.6252971
20. Jia G, Cai Z, Musolesi M, Wang Y, Tennant DA, Weber RJ, Heath JK, He S (2012) Community detection in social and biological networks using differential evolution. In: Learning and

intelligent optimization. Springer, Berlin, pp 71–85. https://doi.org/10.1007/978-3-642-34413-8_6
21. Shang R, Bai J, Jiao L, Jin C (2013) Community detection based on modularity and an improved genetic algorithm. Phys A 392(5):1215–1231. https://doi.org/10.1016/j.physa.2012.11.003
22. Liu D, Jin D, Baquero C, He D, Yang B, Yu Q (2013) Genetic algorithm with a local search strategy for discovering communities in complex networks. Int J Comput Intell Syst 6(2):354–369. https://doi.org/10.1080/18756891.2013.773175
23. Shi C, Cai Y, Fu D, Dong Y, Wu B (2013) A link clustering based overlapping community detection algorithm. Data Knowl Eng 87:394–404. https://doi.org/10.1016/j.datak.2013.05.004
24. Zadeh PM, Kobti Z (2015) A multi-population cultural algorithm for community detection in social networks. Procedia Comput Sci 52:342–349. https://doi.org/10.1016/j.procs.2015.05.105
25. Gupta S, Mittal S, Gupta T, Singhal I, Khatri B, Gupta AK, Kumar N (2017) Parallel quantum-inspired evolutionary algorithms for community detection in social networks. Appl Soft Comput 61:331–353. https://doi.org/10.1016/j.asoc.2017.07.035
26. Zhang L, Pan H, Su Y, Zhang X, Niu Y (2017) A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection. IEEE Trans Cybern 47(9):2703–2716. https://doi.org/10.1109/TCYB.2017.2711038
27. Guerrero M, Montoya FG, Baños R, Alcayde A, Gil C (2017) Adaptive community detection in complex networks using genetic algorithms. Neurocomputing 266:101–113. https://doi.org/10.1016/j.neucom.2017.05.029
28. Wen X, Chen WN, Lin Y, Gu T, Zhang H, Li Y, Yin Y, Zhang J (2017) A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE Trans Evol Comput 21(3):363–377. https://doi.org/10.1109/TEVC.2016.2605501
29. Guendouz M, Amine A, Hamou RM (2017) A discrete modified fireworks algorithm for community detection in complex networks. Appl Intell 46(2):373–385. https://doi.org/10.1007/s10489-016-0840-9
30. Chen J, Wang H, Wang L, Liu W (2016) A dynamic evolutionary clustering perspective: community detection in signed networks by reconstructing neighbor sets. Phys A 447:482–492. https://doi.org/10.1016/j.physa.2015.12.006
31. Ju Y, Zhang S, Ding N, Zeng X, Zhang X (2016) Complex network clustering by a multi-objective evolutionary algorithm based on decomposition and membrane structure. Sci Rep 6:33870. https://doi.org/10.1038/srep33870
32. Rani S, Mehrotra M (2018) A hybrid bat algorithm for community detection in social networks. In: International conference on intelligent systems design and applications. Springer, Cham, pp 943–954. https://doi.org/10.1007/978-3-030-16660-1_92
33. Behera R, Rath S, Misra S, Damaševičius R, Maskeliūnas R (2017) Large scale community detection using a small world model. Appl Sci 7(11):1173. https://doi.org/10.3390/app7111173
34. Ji Z, Pi H, Wei W, Xiong B, Woźniak M, Damasevicius R (2019) Recommendation based on review texts and social communities: a hybrid model. IEEE Access 7:40416–40427. https://doi.org/10.1109/ACCESS.2019.2897586
35. Azaouzi M, Rhouma D, Romdhane LB (2019) Community detection in large-scale social networks: state-of-the-art and future directions. Soc Netw Anal Min 9(1):23. https://doi.org/10.1007/s13278-019-0566-x
36. Moscovici S (1988) Notes towards a description of social representations. Eur J Soc Psychol 18(3):211–250. https://doi.org/10.1002/ejsp.2420180303
37. Pan Y, Li DH, Liu JG, Liang JZ (2010) Detecting community structure in complex networks via node similarity. Phys A 389(14):2849–2857. https://doi.org/10.1016/j.physa.2010.03.006
38. Hunter PR, Gaston MA (1988) Numerical index of the discriminatory ability of typing systems: an application of Simpson's index of diversity. J Clin Microbiol 26(11):2465–2466
39. Hamers L (1989) Similarity measures in scientometric research: the Jaccard index versus Salton's cosine formula. Inf Process Manag 25(3):315–318
40. Blogs network dataset-KONECT, October 2016. https://doi.org/10.1145/1134271.1134277
41. Chicago network dataset-KONECT, October 2016. http://konect.uni-koblenz.de/networks/tntp-ChicagoRegiona
42. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405. https://doi.org/10.1007/s00265-003-0651-y
43. Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. In: Advances in neural information processing systems, pp 539–547. http://dx.doi.org/10.1145/2556612
44. Newman ME (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104. https://doi.org/10.1103/PhysRevE.74.036
45. Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech: Theory Exp 2005(09):P09008
46. Krieger AM, Green PE (1999) A generalized Rand-index method for consensus clustering of separate partitions of the same data base. J Classif 16(1):63–89. https://doi.org/10.1007/s003579900043