Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 3356 – 3367
© TÜBİTAK
doi:10.3906/elk-1902-163

Community detection in complex networks using a new agglomerative approach

Majid ARASTEH, Somayeh ALIZADEH[*]
Faculty of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran,

Received: 25.02.2019 • **Accepted/Published Online:** 08.05.2019 • **Final Version:** 18.09.2019

Abstract: Complex networks are used for the representation of complex systems such as social networks. Graph analysis comprises various tools such as community detection algorithms to uncover hidden data. Community detection aims to detect similar subgroups of networks that have tight interconnections with each other while, there is a sparse connection among different subgroups. In this paper, a greedy and agglomerative approach is proposed to detect communities. The proposed method is fast and often detects high-quality communities. The suggested method has several steps. In the first step, each node is assigned to a separated community. In the second step, a vertex is selected randomly and then its neighbors are determined. Then the selected node and its best neighbor will be merged if their merging brings positive gain. The merging of the selected vertex and its best neighbor has more gain than other neighbors. Whenever the merging occurs, the graph will be updated and this process will be continued until all the vertexes are assessed. Furthermore, the computational complexity of the proposed method is $O(nm)$, where n and m refer to the total number of vertexes and edges, respectively. In addition, our proposed method is compared with the Girvan–Newman algorithm and the fast divisive method for community detection. Results show that the proposed method is much faster than them and can detect high-quality communities. Finally, the accuracy of the proposed method is evaluated by using four different measures of purity, F-measure, NMI, and ARI.

Key words: Complex networks, social networks, community detection, agglomerative approach

1. Introduction

Network analysis has become one of the most important and popular topics in computer science. In mathematics, complex networks are also expressed as graphs. The graph is defined as $G = (V, E)$, where V and E denote vertexes (nodes) and edges (connection), respectively. The graph is used for the representation of data and their relations [1]. Complex networks can be used to represent any complex system such as railways, airlines, and social networks [2–4]. In nonrandom graphs such as social networks, it is expected to detect significant information such as similar groups of communities. Graph mining is a term that is frequently used in graph analyses such as community detection, frequent pattern mining, link analysis, and graph diameter calculation [2]. Graph analysis can be useful as it can assist in identifying a complex system and its patterns in order to provide a hypothesis about the structure of a network [5]. This paper is focused on community detection.

In the literature, a community is also referred to as a group, cluster, or module [6]. In complex systems such as societies, social networks, politics, economics, and many others, it is expected to discover similar communities [4, 7], such as similar groups of researchers at a university. In addition, community detection can help analyzers find the most important groups of people in a society. By the detection of these groups, analyzers can control the propagation of viruses and diseases or rumors in society. Moreover, the analysis of

[*] Correspondence: s_alizadeh@kntu.ac.ir

communities can be used to recommend some special products to a group of people [8]. In social networks, community detection is used for link prediction and the determination of users with similar behaviors [9]. In addition, users with malicious activities can be detected in security systems by using community detection [10]. Finally, information cascade is another important topic in complex network analysis. Information cascade can specify the spreading of diseases, rumors, marketing campaigns, or memes that originally start from a node or set of nodes in a complex network [11].

Furthermore, communities can be hierarchical, having some smaller communities in their structure [12]. For instance, the engineering faculty in a university can be considered as a community that contains some smaller communities such as a computer, mechanical, and electrical engineering communities in its structure. Therefore, community detection aims to detect the groups of a complex network and its hierarchy [12].

Although there is not a special definition for communities, visually the nodes of a community have a dense connection with each other and there are sparse interconnection edges among different communities [6, 13]. Formally, if $G(V,E)$ is supposed as a graph, then G is a dense graph if $E \gg V$.

For the sake of community detection, various methods such as spectral, hierarchical, and partitioning methods are specified. In this paper, we have introduced a quick agglomerative approach to the discovery of communities. The proposed method often detects high-quality communities and the detected communities do not overlap with each other.

In the rest of this paper, Section 2 introduces the related work. Some important modularity functions are explained in Section 3. The proposed algorithm is highlighted in Section 4. Section 5 analyzes the proposed method. Finally, the paper is concluded in Section 6.

2. Related work

This section will overview the current research in the field of community detection. From one point of view, community detection algorithms can discover overlapped communities, while some others are just able to find nonoverlapped communities [14]. In overlapped communities, each node might belong to more than one community at the same time [15]. From the other point of view, community detection algorithms can be classified into two categories, which are agglomerative and divisive approaches [16]. In the agglomerative methods (also called bottom-up), each node is assigned to a community and then according to their similarities they may join each other and form larger communities. In the divisive methods (also called top-down), a graph is considered as a community and it will break down into some smaller ones during a repetitive process [17].

Kernighan–Lin is a preliminary partitioning algorithm which was introduced by Kernighan and Lin in 1970 [18]. This algorithm is quite fast and its complexity is $O(n^2 logn)$ [12]. This algorithm was first introduced to partition a graph into two parts. In 1980 Kernighan-Lin was extended by Suaris and Kedem to partition a graph into more than one cluster [19]. Runtime and needed storage space will be increased by the augmentation of community numbers. Using predefined knowledge such as the number of communities is necessary for this algorithm [12]. In addition, using hierarchical clustering is the other approach for community detection. The definition of a good distance or similarity function such as Euclidean distance is the starting point of this approach [12].

Girvan–Newman (GN) is an important divisive algorithm introduced in 2002 [20, 21]. This method is based on edge centrality and has four important steps, which are as follows [22]: 1) evaluation of edge centralities, which denotes their importance; 2) removal of the most central nodes; 3) reevaluation of the centrality of all remaining edges; 4) repeating steps 2 and 3 while better communities can be detected. Centrality denotes the intermediation of a vertex in the communication of any pair of vertexes [23]. For unweighted graphs, the complexity of this algorithm is $O(m^3)$ [12, 15]. Although the GN algorithm is popular, it is not scalable enough [12, 24]. In addition, [15] introduced a new fast divisive approach for community detection based on edge degree centrality, which is faster than the GN method. The complexity of this algorithm is $O(n^2)$ Moreover, Radicchi et al. in 2004 introduced the edge clustering coefficient (C^g_{ij}), for the evaluation of edge centralities [6]. According to this measure, a smaller number of clustering coefficients denotes a higher edge centrality. In this method, the idea is based on the number of cycles in a graph. It is expected that cycles in a dense cluster are more than in a nondense cluster. In this method, an edge such as ij is considered and then the cycles with a length of g upon the considered edge will be counted (Z^g_{ij}). Then the number of possible cycles based on the considered edge and its neighbor edges is computed (S^g_{ij}) [6]. Finally, the edge clustering coefficient is calculated according to Eq. (1). In this method, usually cycles with a length of 3 or 4 are considered.

$$C^g_{ij} = \frac{Z^g_{ij} + 1}{S^g_{ij}} \tag{1}$$

Furthermore, some algorithms such as the Louvain algorithm are agglomerative [7]. The Louvain algorithm aims to puts similar vertexes into the same community to create a larger community with high modularity. In this method, modularity gain is used to assess the gain of moving a vertex from a community to another one [7]. The complexity of this algorithm is $O(m)$ [12]. Furthermore, Newman proposed another agglomerative method where each vertex is assigned to a separate community. Then these communities alliteratively merge together and form some larger ones. The complexity of this method is $O(n^2)$ [25].

Spectral clustering is another method for community detection [17]. In this method, objects are mapped to a set of points in space and they will be clustered according to eigenvectors of matrices. Afterwards, these points can be clustered by any clustering method, such as k-means [12]. Donath and Hoffmann in 1973 first used eigenvectors of matrices for clustering. In 1973 Fiedler used the second smallest eigenvectors of a Laplacian matrix to compute a bipartite graph [26]. The complexity of this method is $O(n^3)$ [12].

Modularity-based clustering is another type of community detection algorithm. In this method, a modularity evaluation function such as the GN quality function is considered and then any community detection algorithm is applied to increase the value of the modularity function [8]. Greedy techniques such as simulated annealing (SA) are methods used for the optimization problem. This method first was used by Guimera et al. in 2004 for modularity optimization [27]. The implementation consists of two moves, which are the local move and global move. In the local move, a vertex moves randomly from a community to the other ones in order to increase the modularity. The global move consists of a merge and split. The split is employed to separate a graph into more than one subgraph and increase the modularity, while in the merge two subgraphs connect to each other to form a bigger community [12]. In order to escape the local optima, Massen and Doye in 2005 suggested that, instead of swapping the worst vertexes among communities, a vertex from a group of bad vertexes is chosen randomly for swapping between the communities [28]. Since this algorithm depends on some parameters such as the initial temperature, the definition of the exact complexity is not easy [12].

Extremal optimization (EO) is another optimization method that was used by Duch and Arenas in 2005 for modularity optimization [29]. This method is similar to the Kernigan-Lin algorithm, while in this one, it is not necessary to define the number of communities beforehand. In this method, a graph splits into two smaller subgraphs, and in a repetitive process, each subgraph can be split into more than one subgraph again. During the process of the algorithm, each vertex can move from a community to another one in order to increase the modularity. This approach depends on the initialization step and its complexity is $O(n^2 logn)$ [12].

On the other hand, some other communities have an intersection with each other. The aforementioned methods cannot find overlapped communities. In overlapped communities each vertex can belong to more than one community. The clique percolation method (CPM) is an algorithm introduced by Palla et al. [30]. They used the concept of k-clique to find overlapped communities. According to the definition, two k-cliques are adjacent if they have at least one vertex in common. The union of adjacent k-cliques is called a chained k-clique. The k-clique community is the largest connected subgraph obtained from the union of k-cliques. The k-clique communities can share some vertexes and they can be overlapped. In addition, iterative scan (IS) and rank removal (RaRe) are two other methods for overlapped community detection and their computational complexity is $O(n^2)$ [12].

In recent years, metaheuristic algorithms such as evolutionary algorithms have been used for community detection. Evolutionary algorithms consist of some heuristics used for optimization problems. For instance, Tasgin et al. used the genetic algorithm (GA) for community detection [3]. The complexity of their suggested approach is $O(e)$, where e denotes the whole number of edges. Each solution is a vector with length of N that is known as a chromosome and N is the number of vertexes in a graph. In each chromosome, the index of a gene indicates the label of a vertex in a graph and the value of a gene indicates the community of the corresponding vertex. Mutation and cross-over are the two main operators in the GA. In their suggested crossover method, first, two chromosomes are selected and then some of their peer-to-peer genes values are exchanged. In mutation, a chromosome is selected and then the value of a gene is changed to another value (this value should be in the range of the numbers of communities). In this paper, community variance for each node ($CV(i)$) is used as a clean-up function to improve the quality of detected communities. For a vertex such as i, community variance indicates the number of vertexes outside of the community of vertex i that are connected to i. In good communities, for each vertex CV should have a small value. In their proposed method, the GN modularity function is used as a fitness function.

Shang et al. used the GA for community detection [31]. In order to increase the accuracy of community detection, they used default information such as the number of communities, and also the GN modularity function is used as the fitness function. Solutions are coded into some chromosomes and special mutation and crossover operators are used. By the mutations, some genes of a chromosome are changed randomly to another number. By the crossover, two chromosomes are selected and then two-way crossover is performed. Afterwards, SA is executed to look for the optimal points.

Gong et al. introduced multiobjective discrete particle swarm optimization for community detection [32]. In this paper, kernel K-means and ratio cut are used as two objectives and the goal of optimization is their minimization. In this paper, the GN modularity function is used as the fitness function. In their proposed method, each solution is coded into a vector with a length of n, where n is the number of vertexes. Finally, MOPSO is used for optimization. The complexity of the algorithm is $O(n^2)$.

Some other papers have proposed evolutionary algorithms for community detection. For example, Pizzuti used single-objective genetic algorithms for community detection [33]. The memetic algorithm was used by Gong et al. for clustering [34]. MOGA-net [35], MOCD [36], and MOEA/D-net [37] are some other approaches that use multiobjective optimization for community detection.

3. Modularity functions

The quality evaluation of a detected community is the next important task in community detection algorithms. This process can be done by quality functions, which are also called modularity evaluation functions. In this section, some of the important ones will be introduced.

Performance is a quality function that evaluates the correctness of an edge assignment between any nodes of communities. Eq. (2) shows the performance, which is represented by $P(C)$ [12]. In this equation, n denotes the whole number of vertexes in a complex network and the value of performance is $0 \leqslant P(C) \leqslant 1$.

$$P(C) = \frac{|\{i,j\} \in E, C_i = C_j| + |\{i,j\} \notin E, C_i \neq C_j|}{\frac{n(n-1)}{2}} \tag{2}$$

Coverage is another quality function, which is the ratio of the intracluster edges over the whole number of edges in the graph [12]. If the communities are fully disjointed from each other, the coverage will be equal to 1. Eq. (3) shows the coverage, where E is the total number of edges in a graph and E_C^{in} is the number of intracluster edges. In this equation the value of coverage is $0 \leqslant C(C) \leqslant 1$.

$$C(C) = \frac{|E_l^{in}|}{|E|} \tag{3}$$

The GN modularity evaluation function is another important quality function, which is defined as in Eq. (4) [8][12]:

$$Q = \frac{1}{2m} \Sigma_{ij} (A_{ij} - \frac{d_i d_j}{2m} \sigma(C_i, C_j)) \tag{4}$$

In Eq. 4, m denotes the whole number of edges in the graph, and v and w indicate two vertexes in a graph where d_v and d_w show their degrees. A_{vw} clarifies whether v and w are connected or not. If they are connected, the value of A_{vw} will be set to 1; otherwise, it will be set to 0. If v and w are in the same community then $\sigma(C_i, C_j)$ will be set to 1 and otherwise it will be set to 0. According to Eq. (4), Q will be a large number if the graph has dense communities.

Separability is the other measure to assess the goodness of detected communities. According to this measure, a good community is well separated from the rest of the network [38]. This measure is calculated according to Eq. (5), which is the ratio between the number of internal and external edges of a community.

$$S(C) = \frac{|E_l^{in}|}{|E_l^{out}|} \tag{5}$$

Density is the next measure for the quality evaluation of detected communities, which is the ratio between the intracommunity edges and all possible edges of the supposed community [38]. This measure is computed by Eq. (6). In this equation, n_c refers to the total possible edges of a cluster.

$$S(C) = \frac{2|E_l^{in}|}{|n_c(n_c - 1)|} \tag{6}$$

Furthermore, Blondel et al. [7] introduced a modularity gain function to assess removing a vertex such as i from its community and placing it in another community such as j. The modularity gain function is calculated by Eq. (7).

$$\triangle Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \tag{7}$$

In this equation, $\sum_{in}$ refers to the total weights of the edges inside cluster C. In addition, $\sum_{tot}$ indicates the total weights of the edges incident to the nodes of cluster C. k_i is the degree of node i and $k_{i,in}$ shows the sum of the weights of links from node i to the nodes of cluster C. In this equation, m denotes the total weights of the network's links.

This section introduces some modularity functions used to assess the goodness of detected communities. In this paper, we have used the GN modularity function, which is one of the most popular ones.

4. Proposed method

In this paper, we have introduced a new quick agglomerative approach to detect communities. In the proposed method, at the beginning of the algorithm, each node is assigned to a separated community. In the next steps of the algorithm, the communities might join each other and form some larger communities. The steps of the proposed method can be summarized as follows:

1. Each node is assigned to a separated community.
2. A node is selected randomly. The selected node is called the target node.
3. All neighbors of the target node are determined.
4. For each neighbor of a target community, the modularity gain of the target node and its neighbor is computed to assess their merging gain. In this paper, we have used the modularity gain function of Blondel et al., which is as in Eq. (7) [7].
5. The target node will be merged with its best neighbor if the value of the modularity gain is more than zero. Moreover, merging the target node and its best neighbor has higher modularity gain than merging the target node with its other neighbors.
6. If the target node and its best neighbor are merged and form a larger community, then the network will be updated in this step. By the updating process, the target community and neighbor community will be merged into a unique community and the neighbors of the neighbor node will be the neighbors of the target node in the new network.
7. This process will be continued while the remaining communities cannot merge anymore.

Furthermore, the proposed method for community detection has some important features, which are:

- It is a greedy and agglomerative method for community detection.
- It can detect nonoverlapped communities.
- It is a fast method and its computational complexity is $O(nm)$, where n and m are the numbers of the network's vertexes and edges, respectively.

The pseudocode of the proposed method is as follows:

Algorithm 1: The pseudocode of the proposed method.

```
Data: Network $G = (V, E)$
Result: Set of communities $C = \{c_1, c_2, ..., c_n\}$
$C = \{V_1, V_2, V_3, ..., V_n\}$;
for (i=1:n) do
    $best\_Modularity = -1$;
    $target\_Node$ = Select a random vertex;
    $neighbor\_Nodes$ = Select all neighbors of the target Node;
    $best\_Choice = NULL$;
    for (All neighbors of the target node) do
        $new\_Node = target\_node \cup neighbor\_Node$;
        $modularity\_Gain = Modularity(new\_Node)$;
        if ((modularity_Gain > best_Modularity AND modularity_Gain > 0)) then
            $best\_Modularity \leftarrow modularity\_Gain$;
            $best\_Neighbor = this.neighbor$;
            $best\_Choice \leftarrow new\_Node$;
    if $best\_Choice! = NULL$ then
        $target\_Node = best\_Choice$;
        $target\_Node.neighbors \leftarrow target\_Node.neighbors \cup best\_Neighbors.neighbors$;
```

5. Evaluation

This section will assess the quality, accuracy, and required time of the proposed algorithm. For this, the proposed method will be compared with the GN algorithm [20] and the fast divisive method for community detection [15].

5.1. Quality and time

The proposed method is fast and often detects high-quality communities. The neighbors of a vertex in a complex network are limited to a constant number such as c. Hence, as is obvious from the pseudocode, the complexity of the proposed method is $O(n(c+m))$, where n and m denote the number of vertexes and edges in a network, respectively. The algorithm is repeated n times and each iteration updating process is repeated m times. Since c is a constant number, the computational complexity of the proposed algorithm is equal to $O(nm)$ (the graph is implemented with the edge list as its data structure).

In addition, at each step, a vertex is selected randomly and might be merged with its best neighbor (if their merging is beneficial and increases the modularity). Therefore, at the end of the algorithm, similar vertexes will be merged into the same communities and all possible communities will be detected. Moreover, the final output of the proposed algorithm completely depends on the order of nodes selection. In order to assess the quality of detected communities and the speed of the algorithm, some datasets are used, which are summarized in Table 1.

The output of the proposed algorithms depends on the order of nodes visiting. By consideration of this fact, the proposed method is compared with two other algorithms, which are the GN algorithm [20], and the fast divisive method for community detection [15]. Table 2 compares the quality and speed of the detected communities by our proposed method, the GN, and the fast divisive method. In addition, Figure 1 shows

Table 1. The used datasets in this paper

Rows	Datasets	Category	Vertexes no.	Edges no.
1	Zachary karate club	Human social	34	78
2	Contiguous USA	Infrastructure	49	107
3	Dolphins	Animals	62	159
5	Euroroad	Infrastructure	1174	1417

Table 2. The quality and speed of the proposed method, GN, and fast divisive.

Networks	GN		Fast divisive		Proposed method			
	Q	T	Q	T	WQ	MQ	BQ	T
Zachary karate club	0.06	8	0.11	0.17	0.04	0.12	0.15	0.88
Contiguous USA	0.13	17	0.12	0.43	0.13	0.19	0.24	1.2
Dolphins	0.12	42	0.15	0.81	0.09	0.16	0.18	1.8
Jazz musicians	0.04	11206	0.08	1519	0.02	0.08	0.1	85
Euroroad	0.25	132000	0.39	13705	0.25	0.35	0.38	185

- In this table Q and T refer to quality and time, respectively.
- In addition, WQ, MQ, and BQ indicate worst quality, middle quality, and best quality of detected communities.
- Moreover, the values of time are represented in seconds (s).

the quality of detected communities by our proposed method, the GN, and the fast divisive method. This figure shows that our proposed method on average and in some cases detects better communities than the other methods while it is much faster than them. In this figure, the quality of detected communities by each algorithm on some datasets and their required times in seconds are highlighted. Moreover, high speed is the main feature of the proposed method that makes it useful for large-scale network analysis.

5.2. Accuracy

In Section 5.1, the quality of communities detected by three different algorithms and their required times are compared with each other. The results show that the proposed algorithm is faster than the two other methods. In addition, in some cases, the proposed algorithm detects good communities. In this section, the accuracy of the proposed algorithm will be assessed. For the accuracy assessment, the output of an algorithm is compared with the expected results. For this purpose, various measures such as normalized mutual information (NMI), purity, F-measure, and adjusted Rand index (ARI) were introduced [38, 39]. In this section, the accuracy of the proposed method will be compared with the accuracy of the GN and the fast divisive method for community detection. For the accuracy assessment, we have used purity, F-measure, NMI, and ARI, which are given in Eqs. (8), (9), (10), and (11), respectively.

- **Purity**: This is the simplest method for accuracy assessment and it is computed by Eq. (8). In this

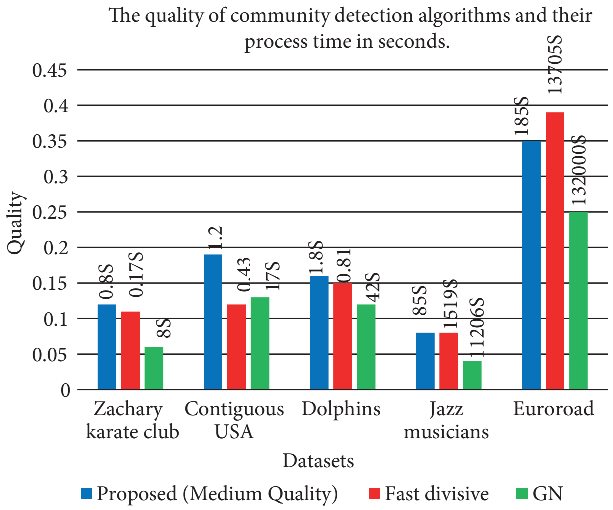

Figure 1. Comparing the quality of detected communities and their running times on five different datasets.

equation, Ω and C refer to the expected and detected communities.

$$Purity(\Omega, C) = \frac{1}{N} \sum_{k} max_j |\Omega_k, C_j| \tag{8}$$

- **F-measure:** This method evaluates the goodness of detected clusters and is as in Eq. (9).

$$F - measure = \frac{2.Purity(\Omega, C).Purity(C, \Omega)}{Purity(\Omega, C) + Purity(C, \Omega)} \tag{9}$$

- **Normalized mutual information (NMI)**: According to this measure, two clusters are compared based on information theory (entropy). Eq. (10) is used for the computation of NMI.

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{l=1}^{k^{(b)}} n_{h,l} log(\frac{n.n_{h,l}}{n_h^{(a)}.n_l^{(b)}})}{\sqrt{(\sum_{h=1}^{k^{(a)}} .n_h^{(a)} log(\frac{n_h^{(a)}}{n}))(\sum_{l=1}^{k^{(b)}} .n_l^{(b)} log(\frac{n_l^{(b)}}{n}))}} \tag{10}$$

- **Adjusted Rand index (ARI):** This is another measure to evaluate the dependencies between two clusters (expected cluster and discovered cluster), according to Eq. (11).

$$ARI = \frac{\Sigma_{i,j}\binom{n_{ij}}{2} - [\Sigma_i\binom{n_i}{2}.\Sigma_j\binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\Sigma_i\binom{n_i}{2} + \Sigma_j\binom{n_j}{2}] - [\Sigma_i\binom{n_i}{2}.\Sigma_j\binom{n_j}{2}]/\binom{n}{2}} \tag{11}$$

Finally, considering the Zachary karate club database, we have compared the accuracy of the proposed method with the accuracy of the GN algorithm and the fast divisive method for community detection. Results show that the proposed method has higher accuracy than the other ones. Figure 2 shows the comparison of accuracy between the GN, fast divisive, and proposed methods by measures of purity, F-measure, NMI, and ARI.

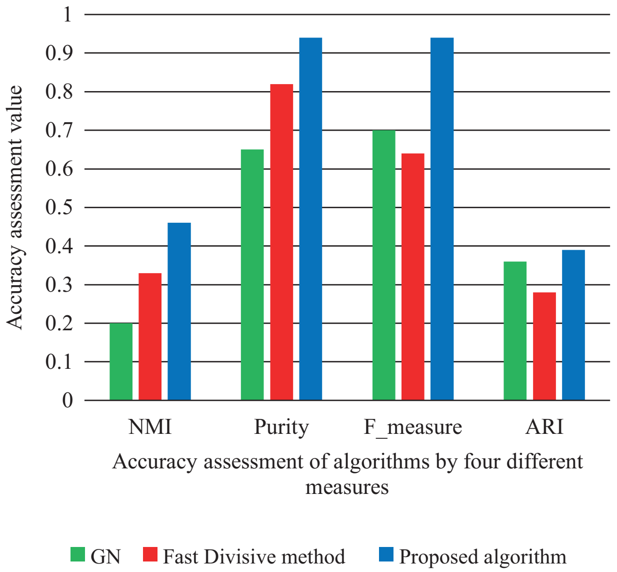

Figure 2. Accuracy assessment of three community detection algorithms by the consideration of four accuracy measures for the Zachary karate club dataset.

6. Conclusion

Many complex systems such as social networks, human societies, and airlines can be represented by complex networks. Complex networks have useful hidden information, which can be discovered by graph analysis methods such as community detection. Community detection aims to discover similar subgroups of nodes in complex networks that have a tight connection with each other, whereas there is a light interconnection between different subgroups. In this paper, a new fast agglomerative community detection algorithm is introduced, which can detect high-quality communities. The proposed methods have several steps. In the first step, all vertexes are assigned to separate communities. In the next step, a node is selected randomly, which is called the target node, and then all neighbors of the target node are determined. Afterwards, the merging gain of the target node and its neighbors are evaluated separately. Then the target node will be merged with its best neighbor if its modularity gain is more than zero. Afterwards, all the changes will be applied to the considered network. This process will be iterated while all the nodes are evaluated. This algorithm is fast and its computational complexity is $O(nm)$, where n and m refer to the total number of vertexes and edges of a network. Afterwards, the proposed method is evaluated by the consideration of its quality, accuracy, and required time for community detection. Results show that the proposed method is much faster than the GN and fast divisive methods for community detection and it can detect good communities. Finally, the accuracy of the proposed method is evaluated by using four measures of purity, F-measure, NMI, and ARI.

References

[1] Mochón MC. Social network analysis and big data tools applied to the systemic risk supervision. International Journal of Interactive Multimedia and Artificial Intelligence 2016; 3 (6): 34-37. doi: 10.9781/ijimai.2016.365

[2] Lee H, Shao B, Kang U. Fast graph mining with HBase. Information Sciences 2015; 315: 56-66. doi:10.1016/j.ins.2015.04.016

[3] Tasgin M, Bingol H. Community detection in complex networks using genetic algorithm. In: Proceedings of the European Conference on Complex Systems; 2006.

[4] Newman ME. Fast algorithm for detecting community structure in networks. Physical Review 2004; 69 (6): 066133. doi: 10.1103/PhysRevE.69.066133

[5] Gosak M, Markovič R, Dolenšek J, Rupnik MS, Marhl M et al. Network science of biological systems at different scales: a review. Physics of Life Reviews 2018; 24: 118-135. doi: 10.1016/j.plrev.2017.11.003

[6] Radicchi F, Castellano C, Cecconi F, Loreto V, Parisi D. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the USA 2004; 101(9): 2658-2663. doi: 10.1073/pnas.0400054101

[7] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008; 2008 (10): 10008. doi:1742-5468/2008/10/P10008

[8] Clauset A, Newman M E, Moore C. Finding community structure in very large networks. Physical Review E 2004; 70 (6): 066111. doi:10.1103/PhysRevE.70.066111

[9] Tan F, Xia Y, Zhu B. Link prediction in complex networks: a mutual information perspective. A mutual information perspective. PLoS One 2014; 9 (9): 107056. doi: 10.1371/journal.pone.0107056

[10] Moghaddam A. Detection of malicious user communities in data networks. MSc, University of Victoria, Victoria, Canada, 2011.

[11] Jalili M, Perc M. Information ascades in complex networks. Journal of Complex Networks 2017; 5 (5): 665-693. doi: 10.1093/comnet/cnx019

[12] Fortunato S. Community detection in graphs. Physics Reports 2010; 486 (3-5): 75-174. doi: 10.1016/j.physrep.2009.11.002

[13] Zeng J, Hongfeng Y. A study of graph partitioning schemes for parallel graph community detection. Parallel Computing 2016; 58: 131-139. doi: 10.1016/j.parco.2016.05.008

[14] Harenberg S, Bello G, Gjeltema L, Ranshous S, Harlalka J et al. Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdisciplinary Reviews: Computational Statistics 2014; 6 (6): 426-439. doi: 10.1002/wics.1319

[15] Arasteh M, Alizadeh S. A fast divisive community detection algorithm based on edge degree betweenness centrality. Applied Intelligence 2019; 49 (2): 689-702. doi: 10.1007/s10489-018-1297-9

[16] Pan Y, Li DH, Liu JG, Liang JZ. Detecting community structure in complex networks via node similarity. Physica A 2010; 389 (14): 2849-2857. doi: 10.1016/j.physa.2010.03.006

[17] Schaeffer SE. Graph clustering. Computer Science Review 2007; 1 (1): 27-64. doi: 10.1016/j.cosrev.2007.05.001

[18] Kernighan BW. An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal 1970; 49 (2): 291-307. doi: 10.1002/j.1538-7305.1970.tb01770.x

[19] Suaris PR, Kedem G. An algorithm for quadrisection and its application to standard cell placement. IEEE Transactions on Circuits and Systems 1988; 35 (3): 294-303. doi: 10.1109/31.1742

[20] Girvan M, Newman ME. Community structure in social and biological networks. Proceedings of the National Academy of Sciences USA 2002; 99 (12): 7821-7826. doi: 10.1073/pnas.122653799

[21] Newman ME. Analysis of weighted networks. Physical Review E 2004; 70 (5): 056131. doi: 10.1103/PhysRevE.70.056131

[22] Newman M, Girvan M. Finding and evaluating community structure in networks. Physical Review E 2004; 69 (2): 026113. doi: 10.1103/PhysRevE.69.026113

[23] Estrada E. Virtual identification of essential proteins within the protein interaction network of yeast. Proteomics 2006; 6 (1): 35-40. doi: 10.1002/pmic.200500209

[24] Hurajová JC, Madaras T. Revising the Newman-Girvan algorithm. In: CEUR Workshop Proceedings of the 16th ITAT Conference Information Technologies - Applications and Theory; Tatranské Matliare, Slovakia; 2016. pp. 200–205.

[25] Newman M. Fast algorithm for detecting community structure in networks. Physical Review E 2004; 69 (6): 066133. doi: 10.1103/PhysRevE.69.066133

[26] Fiedler M. Algebraic connectivity of graphs. Czechoslovak Mathematical Journal 1973; 23 (2): 298-305. doi: dml.cz/dmlcz/101168

[27] Guimera R, Sales-Pardo M, Amaral LAN. Modularity from fluctuations in random graphs and complex networks. Physical Review E 2004; 70 (2): 025101. doi: 10.1103/PhysRevE.70.025101

[28] Massen CP, Doye JP. Identifying communities within energy landscapes. Physical Review E 2005; 71 (4): 046101. doi: 10.1103/PhysRevE.71.046101

[29] Duch J, Arenas A. Community detection in complex networks using extremal optimization. Physical Review E 2005; 72 (2): 027104. doi: 10.1103/PhysRevE.72.027104

[30] Derényi I, Palla G, Vicsek T. Clique percolation in random networks. Physical Review Letters 2005; 94 (16): 160202. doi: 10.1103/PhysRevLett.94.160202

[31] Shang R, Bai J, Jiao L, Jin C. Community detection based on modularity and an improved genetic algorithm. Physica A 2013; 392 (5): 1215-1231. doi: 10.1016/j.physa.2012.11.003

[32] Gong M, Cai Q, Chen X, Ma L. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Transactions on Evolutionary Computation 2014; 18 (1): 82-97. doi: 10.1109/TEVC.2013.2260862

[33] Pizzuti C. GA-Net: A genetic algorithm for community detection in social networks. In: Parallel Problem Solving from Nature (PPSN), Lecture Notes in Computer Science; 2008. pp. 1081-1090. doi:10.1007/978-3-540-87700-4_107

[34] Gong M, Fu B, Jiao L, Du H. Memetic algorithm for community detection in networks. Physical Review E 2011; 84 (5): 056101. doi: 10.1103/PhysRevE.84.056101

[35] Pizzuti C. A multiobjective genetic algorithm to find communities in complex networks. IEEE Transactions on Evolutionary Computation 2012; 16 (3): 418-430. doi: 10.1109/TEVC.2011.2161090

[36] Shi C, Yan Z, Cai Y, Wu B. Multi-objective community detection in complex networks. Applied Soft Computing 2012; 12 (2): 850-859. doi: 10.1016/j.asoc.2011.10.005

[37] Gong M, Ma L, Zhang Q, Jiao L. Community detection in networks by using multiobjective evolutionary algorithm with decomposition. Physica A 2012; 391 (15): 4050-4060. doi: 10.1016/j.physa.2012.03.021

[38] Chakraborty T, Dalmia A. Metrics for community analysis: a survey. ACM Computing Surveys 2017; 50 (4): 54. doi: 10.1145/3091106

[39] Wagner S, Wagner D. Comparing Clusterings: An Overview. Karlsuhe, Germany: Universität Karlsruhe, 2007. doi: 10.5445/IR/1000011477