IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 May 2024, accepted 8 June 2024, date of publication 27 June 2024, date of current version 15 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420109

RESEARCH ARTICLE

Community Detection in Social Networks Using a Local Approach Based on Node Ranking

JAFAR SHEYKHZADEH[1], BAGHER ZAREI[2], AND FARHAD SOLEIMANIAN GHAREHCHOPOGH[1]

[1]Department of Computer Engineering, Islamic Azad University, Urmia Branch, Urmia 57169-63896, Iran
[2]Faculty of Computer Engineering and Information Technology, Islamic Azad University, Shabestar Branch, Shabestar 53816-37181, Iran

Corresponding author: Bagher Zarei (zarei.bager@iau.ac.ir)

ABSTRACT Community detection is crucial for analyzing the structure of social networks and extracting hidden information from them. The goal is to find groups of nodes (communities) with high intra-group and low inter-group communications. This problem is NP-hard, and most existing algorithms are global with high computational complexity, especially for large networks. Recently, local methods with acceptable computational complexity have been developed, but many have low accuracy and are non-deterministic. This paper introduces a new local algorithm, LCD-SN, which identifies communities based on first- and second-degree neighbor nodes. Unlike other local algorithms, LCD-SN is highly accurate, definitive, and not dependent on initial seed nodes. Additionally, a new index is proposed to determine the importance of network nodes using their local characteristics (first- and second-degree neighbors). Using this index, LCD-SN first identifies important nodes, forms initial communities with these nodes and their first-degree neighbors, and then obtains final communities through post-processing. Experiments show that LCD-SN is effective in identifying communities in social networks.

INDEX TERMS Community structure detection, modularity, nodes ranking, social network analysis.

I. INTRODUCTION

Many existing systems can be modeled using graphs. In mathematics, a graph is shown as $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of $|V| = n$ vertices (nodes) and $E \subseteq V \times V$ is a set of $|E| = m$ edges. For example, in a social network graph, nodes show people, and edges show connections between them. All nodes connected to the vertex v_i are called its neighborhood set and are denoted by $\Gamma(v_i)$ [1].

The identification of communities is one of the main problems in the field of social network analysis. Identifying communities makes it possible to extract hidden information in the network and obtain information about people or entities. The community structure detection problem aims to identify groups of nodes so that there are relatively more connections within each group and relatively fewer connections between the groups [2], [3]. So far, many criteria have been proposed to evaluate the structure of the community. One of the most well-known evaluation criteria is the modularity criterion presented by Girvan Newman [4]. The community structure detection problem can be defined as a modularity optimization problem. It has been proven that modularity optimization is an NP-hard problem [2].

The existing algorithms for discovering the community structure are divided into global and local categories. Global algorithms are more accurate, but due to the need to access complete network information, their time complexity is high and cannot be used in large networks. In contrast, local algorithms have less time complexity due to limited information [5], [6]. However, local algorithms have disadvantages, such as their dependence on predefined parameters, instability, and low quality.

In recent years, the use of clustering algorithms to detect communities in social networks has attracted researchers' attention. Due to the lack of access to complete information on social networks, local methods are used to increase efficiency and speed in large-scale networks. In most local methods, communities are determined by selecting important nodes as the core of communities. Most existing works have disadvantages, such as low accuracy due to inappropriate

The associate editor coordinating the review of this manuscript and approving it for publication was Feiqi Deng.

core selection, lack of scalability, and uncertainty in the results. In the proposed algorithm, a new index is proposed to calculate the importance of network nodes based on the structural characteristics of the network. The proposed index reflects the real position of the nodes by considering the first- and second-degree neighbors. After determining the importance of nodes, important nodes are considered as the core of communities. Initial communities are formed around the core nodes based on the structural characteristics of the network. Finally, after post-processing, the final communities are obtained. In addition to having the advantages of existing local methods, the proposed algorithm solves the issues of local methods, including the dependence of communities on core nodes, uncertainty, and low quality. The results of experiments show that the proposed algorithm performs better than the compared algorithms in most cases. The main contribution of this article is as follows:

- A new ranking index called IMP is presented. The IMP index is a local index that determines the importance of nodes based on the first- and second-degree neighborhood.
- A new local algorithm called LCD-SN is presented, which can identify communities with high accuracy using the IMP index.
- The generalized Leicht-Holme-Newman similarity index assigns overlapping nodes to a single community.
- The results of experiments in synthetic and real-world benchmark networks show that the introduced algorithm performs better than other algorithms.

The remainder of this article is organized as follows: Section II reviews some related works. The introduced method is provided in Section III. Section IV includes experimental results and compares the presented method with some known methods. Finally, Section V consists of the conclusion and summary of the article.

II. RELATED WORKS

In a general classification, existing community structure detection algorithms are divided into global and local categories. Global algorithms are more accurate, but due to the need to access complete network information, their time complexity is high and cannot be used in large networks. In contrast, local algorithms have less time complexity due to the use of limited information. However, local algorithms have disadvantages, such as their dependence on predefined parameters, instability, and low quality. The following section has reviewed some of the most well-known global and local algorithms.

A. GLOBAL COMMUNITY STRUCTURE DETECTION ALGORITHMS

The methods based on graph partitioning are among the first global methods for detecting the structure of communities. These methods divide the graph of a social network into k predefined parts (communities) so that the sum of edges between communities is minimized. Graph partitioning methods are inappropriate for discovering communities in large networks due to high computational complexity. The Kernighan-Lin method is one of the most famous algorithms for this category [7].

Hierarchical clustering methods are divided into agglomerative and divisive categories, among other global community detection methods. Divisive hierarchical clustering methods cluster the graph by identifying and removing inter-cluster edges. The algorithm of Girvan-Newman [4], [8] is one of the most popular divisive hierarchical algorithms. This method identifies and removes inter-cluster edges using the edge betweenness centrality. An edge with the highest amount of betweenness is a bridge between communities. This algorithm's time complexity is O(n3). In [9], communities have been identified using the agglomerative hierarchical method based on modularity criterion and cosine similarity index. These methods are only applicable in small networks due to high time complexity.

In [10], a modularity-based heuristic method called Louvain has been introduced. In this method, each node is initially considered an independent community. Then, during an iterative process, those nodes whose merging increases the modularity criterion are merged. The merging of nodes continues until no improvement in modularity is achieved. After forming initial communities, a new graph is created in which each node corresponds to a community of initial communities, and the edge weight between two nodes is equal to the number/sum of the weights of the edges between the two communities corresponding to them. The above steps (merging and forming a new graph) are repeated on the new graph. Merging and creating a new graph continues until no improvement in modularity is achieved.

In [11] and [12], a genetic algorithm named GACD and GATB has been proposed for community discovery. These algorithms are according to modularity optimization. In these algorithms, locus-based adjacency coding is used to represent a graph partition. Locus-based adjacency coding for community detection has the following benefits: (1) its search space is smaller than string coding, (2) the community number is automatically specified in the decoding process, and (3) the decoding process is very efficient. Crossover and mutation operators do not lead to invalid solutions in this coding.

In [13], three algorithms named MEM-net, OMA-net, and GAOMA-net have been introduced to detect community structures in complex networks. These algorithms do not require previous information, such as the community number, and identify communities dynamically. In the GAOMA-net algorithm, which is the main algorithm introduced in this paper, the Object Migration Automata (OMA) and genetic algorithm are combined as a single framework, and the algorithm of MEM-net is utilized as a heuristic to generate a part of the initial population. This combination accelerates the algorithm convergence and prevents it from falling into the local optimum.

In [14], a chaotic memetic algorithm (CMA) has been introduced to identify communities. In the CMA algorithm, a combination of dedicated local search and genetic algorithm (global search) has been utilized to search the solution space. In addition, this article uses chaotic numbers instead of random numbers. Chaotic numbers preserve the diversity of the population and avoid the algorithm falling into the local optimum.

In [15], a chaotic cellular learning automata-based evolutionary model (CCLA-EM) was presented to discover communities in complex networks. CCLA-EM algorithm combines an evolutionary algorithm with cellular learning automata. In this algorithm, the individuals of the population are distributed over the cells of a cellular learning automata. Each individual cooperates and communicates with the individuals of the neighboring cells to achieve the global optimum. Individual distribution over cells in cellular learning automata causes the parallel implementation of the presented model. In addition, this method uses chaotic numbers instead of random numbers. Using chaotic numbers instead of random numbers causes a complete search in the search space and prohibits the algorithm from getting trapped in the local optimum.

B. LOCAL COMMUNITY STRUCTURE DETECTION ALGORITHMS

Label Propagation Algorithm (LPA) [16] is one of the most famous local community discovery algorithms proposed by Raghavan. In this algorithm, first, each node is given a unique label. Then, each node label is updated by majority voting on the labels of its neighbors. The node labels update process continues until no change occurs in the node labels. The most significant advantage of LPA is its linear computational complexity, and one of its disadvantages is its uncertainty. In [17], the problem of indeterminacy and randomness of the LPA has been overcome using the propagation of the label from the least important node.

In [18], using label propagation and MinHash, an algorithm called WLPA has been presented for community detection in unsigned and signed networks. In the WLPA algorithm, communication intensity is also considered in addition to communication. In this method, the similarity of all adjacent nodes is determined using MinHash. Therefore, each edge is given a weight equal to the similarity of its end nodes. The weight given to each edge indicates the intensity of the connection between its end nodes. Finally, the community structure is specified through the propagation of weighted labels. The results of the experiments show that using the WLPA algorithm is effective and efficient for discovering communities in signed and unsigned networks.

In [19], a four-step method has been presented for extracting communities. In the first step, important nodes are found by extracting global and local structures and using game theory. In the second stage, with the propagation of the label, primary communities are formed. In the third stage, the obtained communities are integrated. Finally, in the fourth step, the final communities are extracted by ensuring the correct allocation of nodes. This method has low accuracy in detecting communities in real-world networks.

In [20], a three-step method has been presented for local community detection in social networks. These three steps include (1) determining the importance of nodes and forming primary communities, (2) extending primary communities, and (3) merging primary communities to obtain final communities. In the first stage, the importance of network nodes is calculated using an index called RDC. Then, initial communities are formed, including several important nodes. In the second stage, the initial communities are extended using a local similarity measure so that each network node becomes a community member. The extracted communities are investigated in the third stage, and small communities are merged with other communities.

In [21], a two-stage local community discovery algorithm, RTLCD, based on community expansion and core detection, has been presented. Local community discovery algorithms have two common challenges and issues: (1) seed-dependent problem, meaning whether the seed node location affects the quality of the detected local community or not, and (2) the invalid core problem, which means that some local algorithms cannot ensure the core node participates in the final community. This article used the criteria of node relation strength and node mass (node similarity and local indices for node centrality), as well as community relationship strength, to solve the two mentioned problems.

In [22], a three-stage local community detection algorithm called Three-Stage (TS) has been presented based on global and local information. These steps include central node recognition, label propagation, and community integration. Central nodes are determined based on the distance between them (greater than the average distance). Label propagation means labeling nodes with identical colors when they reach maximum similarity. The integration of communities is done if the modularity increase is positive and at its maximum level.

In [23], a local three-stage algorithm called LCDPC has been presented for community detection, which identifies local communities based on exploring potential communities. In this algorithm, first, an appropriate node is found to replace a particular node as a seed by determining the similarity and importance of the node. Then, the initial community is formed by the composition of the seed and its appropriate potential community. Eventually, the eligible nodes are chosen by comparing the similarity between expanding nodes and potential communities to add to the initial community. This algorithm has a good speed in detecting communities, but the quality of the extracted communities is similar to other algorithms.

In [24], a novel approach to community discovery is proposed by considering each node's importance score and membership degree. Nodes with higher importance scores can

become the core of communities, forming communities with a reasonable number of nodes.

In [25], a group clustering algorithm based on influential nodes has been introduced to improve the detection of communities in complex networks. Considering the diverse attributes of the network, the proposed approach searches to find shared interests among users and their behaviors to identify the most appropriate communities. Initially, a group of influential nodes are recognized as community centers. Then, initial communities are created based on the determined centers. Finally, the initial communities are reclustered to form the final communities.

In [26], a novel approach called Deep Semi-Supervised Community Detection (DSSC) has been presented for complex network clustering. DSSC utilizes a semi-autoencoder (SeAE) with a pair-wise constraint matrix based on point-wise mutual information (PMI) to learn distinctive features and enhance clustering accuracy.

In [27], a new method called modified DeepWalk has been proposed by combining deep learning and random walk. It employs a modified Random Walk with Restart to capture k-order structural and attribute information, enabling the modeling of interactions between network structure and high-order proximities.

In [28], an extension of Symmetric Nonnegative Matrix Factorization (SNMF) called Weighted Symmetric Nonnegative Matrix Factorization (WSNMF) is designed to address challenges encountered in attributed graph clustering. While SNMF has shown promise in graph clustering, it lacks consideration for attributed information, geometric data point structures, and the ability to discriminate irrelevant features and outliers.

In [29], a rapid and precise community detection method in large networks based on locally balanced label propagation is proposed. A novel local similarity measure is also presented to assign importance to nodes. This method demonstrates fast convergence speed along with stable and accurate results.

III. THE PROPOSED ALGORITHM

In this section, a new index called IMP has been presented to determine the importance of nodes. Then, using the IMP index, a local algorithm called LCD-SN has been presented to determine communities in social networks. The flowchart of the proposed method is illustrated in Figure 1. In addition, the notations used in presenting the proposed algorithm are listed in Table 1.

A. IMP IMPORTANCE INDEX

This subsection presents a new index called IMP to determine the importance of nodes according to the nodes' first- and second-degree neighbors. In the IMP index, structural and local characteristics of nodes have been used to determine their importance. The law of continuity of electric current in electric circuits inspires this index. According to the continuity law of electric current, the totality of currents entering a node equals the totality of currents leaving that node. In other words, for each node of an electric circuit, we have

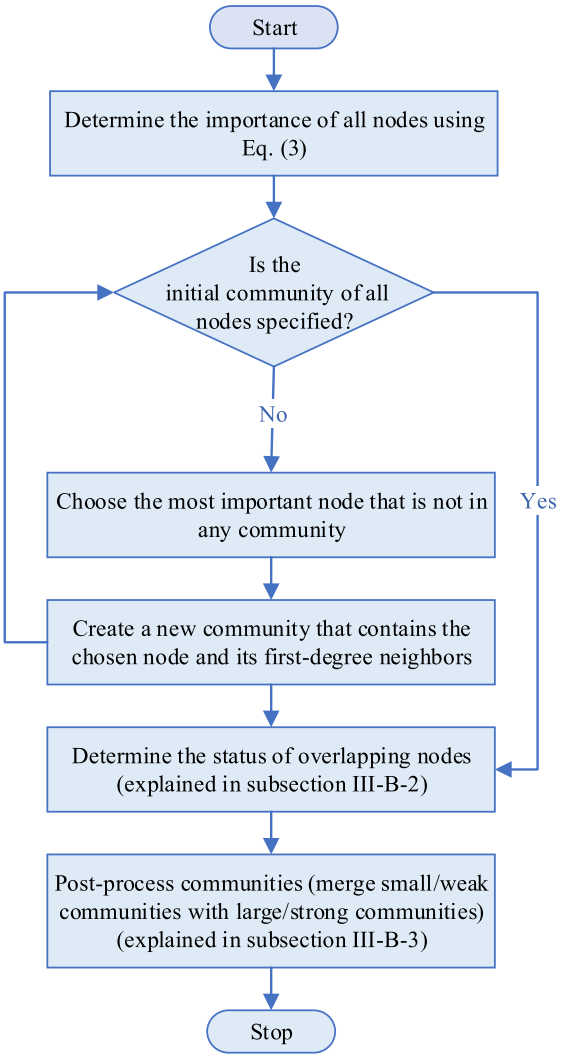

FIGURE 1. Flowchart of the LCD-SN algorithm.

$$\sum I^{in} = \sum I^{out} \tag{1}$$

The graph of a directed and weighted social network can be considered as corresponding to an electrical circuit where (a) each of its nodes corresponds to a node of the electrical circuit and (b) the importance of a node is equal to the sum of the weighted incoming/outgoing electric currents to/from that node. Considering that in real-world networks, the importance of a node is affected by the nodes that have a link to it and the intensity of the link (connection weight) between the nodes, Eq. (2) can be used to determine the importance of a node

$$IMP\,(i) = \sum_{j \in \Gamma_i^{in}} \frac{W_{ji} \times IMP(j)}{W_j^{out}} \tag{2}$$

In Eq. (2), Γ_i^{in} is the set of nodes linked to i, W_{ji} is the directed edge weight from node j to node i, and W_j^{out} is the sum of the outgoing edge weight from node j. For example, consider

TABLE 1. Notations used in the proposed algorithm.

Notation	Description
$IMP(i)$	Importance of node i
Γ_i^{in}	The set of nodes linked to i
W_{ji}	The directed edge weight from node j to node i
W_j^{out}	The sum of the weight of outgoing edges from node j
α	Influence of the first-degree neighbors on the importance of a node
β	Influence of the second-degree neighbors on the importance of a node
γ	The maximum number of iterations for calculating the importance of nodes
$N_i^{1,2}$	The set of first- and second-degree neighbors of node i
C_i	i^{th} community
E_i^{in}	Number of edges inside i^{th} community
E_i^{out}	Number of edges from i^{th} community to other communities
m_c	The integration coefficient of the communities
A_{ij}	The element of i^{th} row and j^{th} column of the adjacency matrix A
P_{ij}	The expected value of the number of edges between two nodes i and j

Figure 2, which represents a portion of a social network graph. Suppose the importance of nodes x, y, and z equals 10, 20, and 30, respectively. According to Eq. (2), the importance of node i is equal to

$$IMP(i) = \frac{2 \times 10}{5} + \frac{1 \times 20}{2} + \frac{2 \times 30}{6} = 24$$

Considering that we initially do not know the importance of nodes, the initial importance of all nodes is assumed to be the same. There may also be nodes whose importance is mutually dependent (nodes i and x in Figure 2). Therefore, calculating the node's importance is repeated until either the importance of the nodes converges or the process of calculating the node's importance is repeated a maximum of γ times. The Eq. (2) is defined only according to the first-degree neighbors of the nodes. To increase the accuracy, in addition to the first-degree neighbors, second-degree neighbors can also be considered. In this case, Eq. (2) will be generalized as

$$IMP(i) = \sum_{j \in \Gamma_i^{in}} \left[\alpha \frac{W_{ji} \times IMP(j)}{W_j^{out}} + \sum_{k \in \Gamma_j^{in}} \beta \frac{W_{kj} \times IMP(k)}{W_k^{out}} \right] \quad (3)$$

In Eq. (3), α and β indicate the influence of first- and second-degree neighbors on the importance of node i and Γ_i^{in} is the set of nodes linked to i.

In Table 2, using IMP, PageRank [30], and Degree [31] indices, the importance of the Figure 3 network nodes has been presented. As observed, nodes 4, 12, 2, 9, and 11 were the five nodes with the highest rank in all three indices. This shows that the IMP index aligns with other important indices in the literature, such as PageRank and Degree index. Nodes 4, 12, and 2 (the three nodes with the highest rank using the IMP index) could be used as seeds in the proposed algorithm to form initial communities.

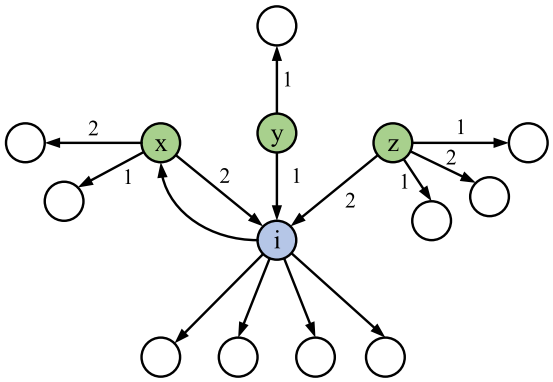

FIGURE 2. A part of the graph of a directed and weighted social network. According to Eq. (2), the importance of node *i* depends on the importance of nodes *x*, *y* comma and *z*.

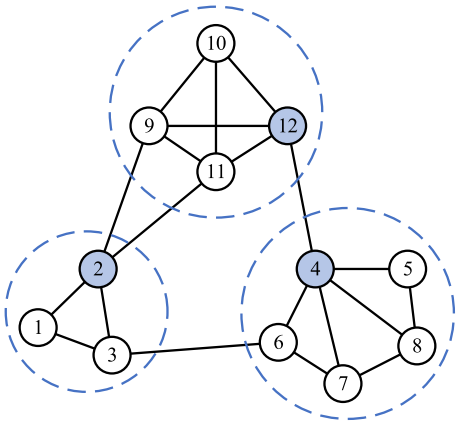

FIGURE 3. An example network having 12 nodes and three communities.

TABLE 2. Ranking of network nodes in Figure 3 using IMP, PageRank, and degree criteria.

IMP		PageRank		degree	
Rank	Value	Rank	Value	Rank	Value
4	2.5250	4	0.1227	4	5
12	2.1233	2	0.0985	2	4
2	2.0833	9	0.0936	9	4
9	2.0183	11	0.0936	11	4
11	2.0183	12	0.0935	12	4
6	1.7167	3	0.0793	3	3
7	1.7017	8	0.0792	6	3
8	1.6683	6	0.0779	7	3
3	1.5933	7	0.0779	8	3
10	1.4850	10	0.0721	10	3
5	1.2083	1	0.0559	1	2
1	1.1333	5	0.0558	5	2

B. LCD-SN ALGORITHM FOR COMMUNITY STRUCTURE DETECTION

In this subsection, an algorithm called LCD-SN has been presented to discover community structure in social networks. The LCD-SN algorithm consists of three phases: (1) forming initial communities, (2) determining the status of overlapping

nodes, and (3) merging communities. In the following, each of these phases has been explained in detail.

1) THE FIRST PHASE: FORMATION OF INITIAL COMMUNITIES

In this phase, the network nodes are first ranked using the Eq. (3). Then, initial communities are formed using the following procedure

(1) The node with the highest rank value, which is not already in any community, is selected as the community's core. Suppose the selected node is k.
(2) Node k, along with its first-degree neighbors, is considered a community.
(3) Steps 1 and 2 are repeated until the initial community of all nodes is determined.

In this phase, some nodes may be members of several communities. In other words, initial communities may overlap. By applying this phase on the Karate network (Figure 4(a)), four initial communities C1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 18, 20, 22, 32}, C2 = {34, 9, 10, 14, 15, 16, 19, 20, 21, 23, 24, 27, 28, 29, 30, 31, 32, 33}, C3 = {26, 24, 25, 32}, and C4 = {17, 6, 7} are obtained. These communities are shown in various colors in Figure 4(b). Note that nodes 6, 7, 9, 14, 20, 24, and 32 overlap and are shown in black.

2) SECOND PHASE: DETERMINING THE STATUS OF OVERLAPPING NODES

In this phase, overlapping nodes are placed only in one of the communities and are removed from the rest of the communities. For this purpose, the similarity of each overlapping node with all the communities to which it belongs is calculated. The overlapping node is placed within the community with the most similarity. In this article, the Generalized Leicht-Holme-Newman (GLHN) similarity index [32] is used to calculate the similarity of a node to a community. In GLHN, to identify the similarity of two nodes, in addition to the first-degree neighborhood, their second-degree neighborhood is also considered. This index has been given in Eq. (4).

$$GLHN\ (i,j) = \frac{\left|N_i^{1,2} \cap N_j^{1,2}\right|}{\left|N_i^{1,2}\right| * \left|N_j^{1,2}\right|} \tag{4}$$

In Eq. (4), $N_i^{1,2}$ represents the set of first- and second-degree neighbors of node i. Using Eq. (4), the similarity of node i to community C is defined as Eq. (5).

$$sim\ (i,C) = \sum_{\substack{j \in C,\\ j \in \Gamma_i}} GLHN\ (i,j) \tag{5}$$

By applying this phase to the output of the first phase (Figure 4(b)), the state of the nodes and communities of Zachary's karate club will be as shown in Figure 4(c). As observed, overlapping nodes 6,7,14, and 20 are assigned to community $C1$, and the rest (nodes 9, 24, and 32) are assigned to community $C2$.

3) THE THIRD PHASE: INTEGRATION OF COMMUNITIES

Considering that some of the communities obtained in the previous phase may be too small or weak, in this phase, the communities obtained in the previous phase are merged in two stages so that the final communities are obtained: (1) the integration of small communities with large communities, and (2) the integration of weak communities with strong communities. In merging small communities with large communities, first, small communities (communities with less than three nodes) are identified. Then, each node of them is transferred to one of the large neighboring communities that is most similar to it. It should be noted that Eq. (5) is used to calculate the similarity of a node to a community.

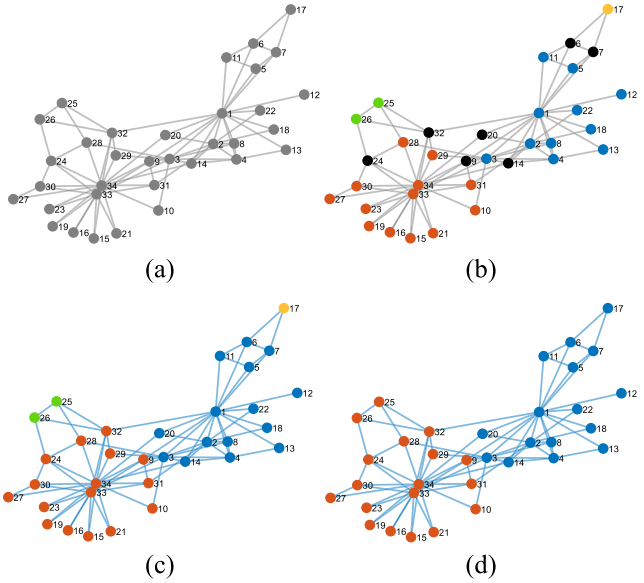

FIGURE 4. (a) Zachary's karate club network and (b)/(c)/(d) the result of the first/second/third phase of the proposed algorithm on it.

Some extracted communities may not be of good quality. To increase the quality of communities, weak communities should be integrated with strong communities. In this paper, a group of nodes is considered a weak community if the condition $E_i^{in} \leq m_c \times E_i^{out}$ is satisfied. In this equation, E_i^{in}, E_i^{out}, and m_c, respectively, indicate the edge number inside the community, the edge number outside the community, and the integration coefficient of the communities. The process of merging weak communities with strong communities continues until there are no weak communities. A weak community's nodes are transferred to a strong neighboring community that are most similar to it. It should be noted that two communities are said to be neighbors if there is at least one edge between them. We can use Eq. (6), a generalization of Eq. (5), to calculate the similarity of the two communities. It should be noted that integration is carried out if it leads to increased modularity. Based on the experimental results, by choosing $m_c = 4$, the final community structure will have

Algorithm 1 LCD-SN (Local Community Detection in Social Networks)

Input: G: Social Network Graph
α: Influence of first-degree neighbors on the importance of a node
β: Influence of second-degree neighbors on the importance of a node
γ: Maximum number of iterations for calculating the importance of a node
Output: Community structure of the input graph G
// First Phase: Formation of Initial Communities
1 Calculate the importance of all nodes of the input graph G using Eq. 3;
2 **repeat**
3 Choose the most important node that is not in any community;
4 Create a new community that contains the chosen node and its first-degree neighbors;
5 **until** *the initial community of all the nodes is determined*;
// Second Phase: Determining Status of Overlapping Nodes
6 **foreach** *overlapping node v* **do**
7 Calculate the similarity of node v with all the communities it belongs to using Eq. 5;
8 Place node v in the community that is most similar to it;
// Third Phase: Integration of Communities
// Merge small communities with large ones
9 $L \leftarrow$ nodes of small communities (communities with less than 3 nodes);
10 **repeat**
11 $v \leftarrow$ Select a node from list L;
12 Merge node v with the neighboring community that is most similar to it (Calculate the similarity of a node to a community using Eq. 5);
13 **until***list L is empty*;
//Merge weak communities with strong ones
14 **foreach***community* C_i **do**
15 **if** $E_i^{in} \leq m_c \times E_i^{out}$*($C_i$ is a weak community)* **then**
16 Move the nodes of community C_i to the neighboring community that is most similar to it (Calculate the similarity of two communities using Eq. 6);

a favorable quality.

$$sim\,(C_1, C_2) = \sum_{\substack{i \in C_1, \\ j \in C_2, j \in \Gamma_i}} GLHN\,(i, j) \qquad (6)$$

By applying this phase to the output of the second phase of the proposed algorithm (Figure 4(c)), the only remaining member of community $C4$ (node 17) is merged with community $C1$, and the remaining nodes of community $C3$ (nodes 25 and 26) are merged with community $C2$. It should be noted that two large communities will be formed after the integration of small communities (Figure 4(d)), none of which could meet the condition of a weak community, so the third phase is completed. The pseudocode of the proposed algorithm is given in Algorithm 1.

C. COMPUTATIONAL COMPLEXITY ANALYSIS

The LCD-SN algorithm consists of three phases: (1) the formation of initial communities, (2) determining the status of overlapping nodes, and (3) the integration of communities. Suppose n indicates the number of network nodes and k indicates the average degree of nodes. Also, we know that the time complexity of calculating the similarity between all pairs of connected nodes using the GLHN index is $\theta\left(nk^3 \log_2 k\right)$. The time complexity of the first phase is $n\left(k^2 + k^3\right) + n \log_2 n + nk \in O\left(nk^3\right)$, where $n\left(k^2 + k^3\right)$ is the time complexity of calculating the importance of nodes, $n \log_2 n$ is the time complexity of sorting nodes' importance, and nk is the time complexity of forming initial communities (lines 1 to 5 of Algorithm 1). The time complexity of the second phase is $O\left(nk^3 \log_2 k\right)$ (lines 6 to 8 of Algorithm 1). The time complexity of the third phase is $nk^3 \log_2 k + nk^3 \log_2 k \in O\left(nk^3 \log_2 k\right)$, where the first term is the time complexity of merging small communities with large ones, and the second term is the time complexity of merging weak communities with strong ones (lines 9 to 16 of Algorithm 1). Finally, the total time complexity of the LCD-SN algorithm is $O\left(nk^3\right) + O\left(nk^3 \log_2 k\right) + O\left(nk^3 \log_2 k\right) \in O\left(nk^3 \log_2 k\right)$. The degree distribution follows the power-law in scale-free networks such as social networks. This means that most nodes have a relatively low degree. Therefore, k will be a small constant value. Thus, the time complexity of the LCD-SN is approximately $O(n)$, which is linear in terms of n.

TABLE 3. Parameters used in different algorithms.

Algorithm	Parameter	Value
CMA [14]	Population size	100
	Crossover rate	0.8
	Mutation Rate	0.2
	Elitism rate	0.05
	Max generations	100
	Local search rate	0.1
	Logistic map parameter	4
GAOMA-net [33]	Population size	100
	Recombination rate	0.8
	Mutation rate	0.2
	Elitism rate	0.05
	Max generations	100
	Object migrating automaton type	Tsetlin
	Actions memory depth	5
GACD [11]	Population size	100
	Crossover rate	0.8
	Mutation rate	0.2
	Max generations	100
GATB [12]	Population size	100
	Crossover rate	0.8
	Mutation rate	0.2
	Initial rate	0.6
	Elitism rate	0.1
	Clean rate	0.5
	Clean threshold	0.7
	Max generations	100
LabelRank [34]	Inflation power	4
	Cutoff threshold	0.1
	Conditional update coefficient	0.7
	Max Iterations	50
LBLD [29]	-	-
CSLPR [17]	-	-
LCD-SN	α	0.7
	β	0.3
	γ	6

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the efficiency of the introduced algorithm is evaluated on real-world and synthetic benchmark networks and is compared with CMA [14], GAOMA-net [33], GACD [11], GATB [12], LabelRank [34], LBLD [29], and CSLPR [17] algorithms.

A. PARAMETER SETTING

The proposed and compared algorithms are implemented using MATLAB R2022a and run on a personal computer with a core i9-13900K 3.00 GHz processor and 64GB of RAM. Table 3 gives the values of parameters used in different algorithms.

B. EVALUATION CRITERIA

To evaluate and compare the algorithms, two measures of Q (modularity) [4] and NMI (Normalized Mutual Information) [35] are utilized. The modularity criterion is used when the ground-truth community structure is not attainable, and the NMI criterion is utilized when the structure of the ground-truth community is available. Modularity is a measure for calculating the quality of dividing nodes into different communities. Due to its simplicity and effectiveness, this criterion has become the most widely used quantitative criterion for comparing different community structure detection algorithms. The modularity measure compares the number of edges within the communities with the null model. The null model is a random graph (edges are located randomly among the nodes) with the same size and degree distribution as the desired graph. If the edge number within the communities of the found community structure is more than the edge number of the corresponding clusters in the null model, the modularity value will be positive and, otherwise, negative. The modularity criterion is defined as Eq. (7).

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - P_{ij})\delta(C_i.C_j) \tag{7}$$

where m is the number of edges, A is the adjacency matrix, $P_{ij} = \frac{k_i k_j}{2m}$ is the expected value of the number of edges between two nodes i and j, C_i is the community of node i, and the function δ is defined as Eq. (8).

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases} \tag{8}$$

The modularity criterion is in the range of $[-0.5,1]$, and larger values show that the edge density within communities is higher than the null model.

The NMI criterion is based on the information theory, which compares the community's quality in terms of its compatibility with the ground-truth community. Assuming to have the found community structure $A = \{C_1, C_2, \ldots, C_A\}$, and the ground-truth community structure $B = \{C'_1, C'_2, \ldots, C'_B\}$, the NMI criterion then will be defined as Eq. (9).

$$\mathrm{NMI}(A, B) = \frac{-2\sum_{i=1}^{|A|}\sum_{j=1}^{|B|} n_{ij}\log\left(\frac{n_{ij}n}{n_{i.}n_{.j}}\right)}{\sum_{i=1}^{|A|} n_{i.}\log\left(\frac{n_{i.}}{n}\right) + \sum_{j=1}^{|B|} n_{.j}\log\left(\frac{n_{.j}}{n}\right)} \tag{9}$$

The NMI measure is based on the confusion matrix. The confusion matrix rows are related to the ground-truth communities, and its columns are related to the found communities. The element n_{ij} of the confusion matrix shows the number of common nodes in the ground-truth community i and the found community j, and $n_{i.}(n_{.j})$ represents the sum of the i row (j column) elements of the confusion matrix. The NMI criterion is in the range [0,1]. Larger values show that the found community structure is more consistent with the ground-truth community structure.

C. DETERMINING THE OPTIMAL VALUE α, β, AND γ PARAMETERS

In this subsection, to determine the optimal value of α, β, and γ parameters, the proposed algorithm with different combinations $\alpha = \{0.1, 0.2, \ldots, 1\}$, $\beta = 1 - \alpha$ and $\gamma = \{1, 2, \ldots, 10\}$ in Eq. (3) has been applied to various real-world and synthetic benchmark networks. The results of the experiments show that the introduced algorithm performs better for values of $\alpha = 0.7$, $\beta = 0.3$, and $\gamma = 6$.

TABLE 4. The results obtained from the proposed algorithm for the Q criterion in real-world benchmark networks for $\alpha = 0.7$, $\beta = 0.3$, and different values of the γ parameter.

Network	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 6$	$\gamma = 7$	$\gamma = 8$	$\gamma = 9$	$\gamma = 10$
Karete	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**	**0.3715**
Dolphins	0.5087	**0.5187**	0.5005	0.5005	0.5005	0.5005	0.5005	0.5005	0.5005	0.5005
PolBooks	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**	**0.5240**
Football	0.5962	0.5815	**0.6056**	**0.6056**	**0.6056**	**0.6056**	**0.6056**	**0.6056**	**0.6056**	**0.6056**
SFI	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**	**0.6918**
NetScience	0.8434	0.8432	**0.8435**	**0.8435**	**0.8435**	**0.8435**	0.8375	0.8375	0.8375	0.8375
Email	0.5257	0.5297	0.5279	0.5279	0.5266	0.5284	0.5258	**0.5354**	**0.5354**	**0.5354**
PowerGrid	0.9146	0.9101	0.9123	0.9140	0.9137	**0.9159**	0.9122	0.9124	0.9114	0.9107
PGP	0.8584	0.8578	0.8588	0.8626	0.8625	**0.8629**	0.8624	0.8619	0.8621	0.8620
GrQc	0.8270	0.8244	0.8252	0.8246	0.8257	**0.8277**	0.8252	0.8268	0.8258	0.8250
ca-AstroPh	0.5567	0.5574	0.549	0.5542	0.5527	**0.5629**	0.5563	0.5560	0.5557	0.5557
ca-HepTh	0.7255	0.7289	0.7330	0.7311	0.7287	0.7331	0.7323	0.7269	**0.7340**	0.7333
ca-HepPh	0.6074	0.6124	0.6052	0.6085	0.6090	**0.6138**	0.6129	0.6130	0.6130	0.6130
Condmat-2003	0.6821	**0.6872**	0.6807	0.6809	0.6857	0.6776	0.6758	0.6751	0.6785	0.6780
Condmat-2005	0.6425	0.6315	0.6385	**0.6441**	0.6382	0.6311	0.6276	0.6236	0.6176	0.6277
Email Enron	0.5349	**0.5466**	0.5352	0.5390	0.5299	0.5244	0.5132	0.5351	0.5357	0.5353
Collaboration	0.7914	0.7944	0.7995	0.8012	**0.8027**	0.7967	0.7972	0.8011	0.7982	0.7948
Internet	**0.5497**	0.5315	0.5010	0.5004	0.4983	0.4981	0.4981	0.4981	0.4981	0.4981

TABLE 5. The results obtained from the proposed algorithm for the NMI criterion in LFR synthetic benchmark networks for $\alpha = 0.7$, $\beta = 0.3$, and different values of the γ parameter.

μ	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$	$\gamma = 6$	$\gamma = 7$	$\gamma = 8$	$\gamma = 9$	$\gamma = 10$
0.00	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
0.05	0.9902	0.9920	0.9914	0.9909	0.9914	0.9922	0.9915	0.9915	0.9922	**0.9928**
0.10	**0.9831**	0.9814	0.9794	0.9788	0.9798	0.9805	0.9800	0.9800	0.9783	0.9783
0.15	0.9652	0.9698	0.9691	0.9697	0.9738	**0.9744**	**0.9744**	**0.9744**	0.9729	0.9729
0.20	0.9520	0.9535	0.9519	0.9552	0.9561	**0.9569**	**0.9569**	**0.9569**	**0.9569**	**0.9569**
0.25	**0.9278**	0.9278	0.9258	0.9257	0.9257	0.9257	0.9257	0.9257	0.9257	0.9257
0.30	0.9044	0.9067	0.9067	**0.9078**	**0.9078**	**0.9078**	**0.9078**	**0.9078**	**0.9078**	**0.9078**
0.35	0.8352	0.8473	0.8462	0.8485	**0.8487**	**0.8487**	**0.8487**	**0.8487**	**0.8487**	**0.8487**
0.40	**0.7494**	0.7388	0.7340	0.7346	0.7345	0.7345	0.7345	0.7345	0.7345	0.7345
0.45	0.6147	0.6098	0.6168	**0.6196**	**0.6196**	**0.6196**	**0.6196**	**0.6196**	**0.6196**	**0.6196**
0.50	0.4678	0.4556	**0.4725**	0.4566	0.4538	0.4533	0.4533	0.4533	0.4533	0.4533

Therefore, in the results stated in the rest of this article, the values of 0.7, 0.3, and 6 have been considered for α, β, and γ parameters, respectively. For example, in Tables 4 and 5, the results of the introduced algorithm for $\alpha = 0.7$, $\beta = 0.3$, and various values of the γ parameter are given in real-world and synthetic benchmark networks, respectively. As you observe, the introduced algorithm performs better when $\gamma = 6$.

D. EVALUATION WITH REAL-WORLD BENCHMARK NETWORKS

In this section, the efficiency of the introduced algorithm is evaluated in real-world benchmark networks and is compared with CMA [14], GAOMA-net [13], GACD [11], GATB [12], LabelRank [34], LBLD [29], and CSLPR [17] algorithms. The information on the real-world benchmark networks is given in Table 6. The Q criterion is used to evaluate and compare different algorithms because the ground-truth community structure is not attainable in most real-world networks. The average modularity obtained from different algorithms is given in Table 7. In the non-deterministic algorithms (CMA, GAOMA-net, GACD, and GATB), each data is the average of 10 runs. The largest modularity value obtained for each network is written in bold. The LCD-SN has obtained the highest modularity value in PolBooks, Football, PowerGrid, PGP, GrQc, AstroPh, ca-HepTh, ca-HepPh, and Collaboration networks. CMA in the Dolphins and Internet networks, GAOMA-net in the Karete network, GACD in the Karate and SFI networks, LBLD in the NetScience, Email, Condmat 2003, and Condmat 2005 networks, CSLPR in the Email Enron network have obtained the highest modularity value.

In addition, Table 8 shows the results of the Friedman test for ranking different algorithms in real-world networks. As you can see, the Friedman test statistic value (p-value = 0.001) is less than 0.05. Therefore, with a probability of 0.95,

TABLE 6. Real-world benchmark network information. These networks can be downloaded from https://networkrepository.com, https://snap.stanford.edu/data, http://konect.cc/networks, and https://www.cise.ufl.edu/research/sparse/matrices/list_by_id.html websites.

Network	Description	Node	Edge
Karate	Zachary's karate club network [36]	34	78
Dolphins	Dolphins social network [33]	62	159
PolBooks	Books about US politics [37]	105	441
Football	American college football network [8]	115	613
SFI	Santa Fe Institute network [8]	118	200
NetScience	Co-authorship of scientists in network theory [38]	379	914
Email	URV email network [39]	1,133	5,451
PowerGrid	US power grid network [40]	4,941	6,594
PGP	The LCC in the graph of PGP users [41]	10,680	24,316
GrQc	Collaboration network of Arxiv General Relativity [42]	5,242	14,496
ca-AstroPh	Collaboration network of Arxiv Astro Physics [42]	18,772	198,110
ca-HepTh	Collaboration network of Arxiv High Energy Physics Theory [42]	9,877	25,998
ca-HepPh	Arxiv High Energy Physics paper citation network [43]	34,546	421,578
Condmat-2003	Collaboration network, preprints in condensed matter archive [44]	31,163	240,058
Condmat-2005	Collaboration network, preprints in condensed matter archive [45]	40,421	351,382
Email Enron	Email communication network from Enron [46]	36,692	183,831
Collaboration	Collaboration network, preprints in high-energy physics [47]	8,361	15,751
Internet	Structure of Internet routers as of July 22, 2006 [48]	22,963	96,872

TABLE 7. Average obtained modularity by different algorithms in real-world benchmark network.

Network	CMA	GAOMA-net	GACD	GATB	LabelRank	LBLD	CSLPR	LCD-SN
Karate	0.4188	**0.4198**	**0.4198**	0.4127	0.3600	0.3710	0.3715	0.3715
Dolphins	**0.5272**	0.5267	0.5265	0.5263	0.3735	0.3780	0.4780	0.5005
PolBooks	0.5266	0.5222	0.5265	0.5265	0.4946	0.4560	0.4990	**0.5270**
Football	0.5908	0.5936	0.5602	0.5836	0.5918	0.5800	0.5860	**0.6056**
SFI	0.7482	0.7489	**0.7503**	0.7428	0.7227	0.7200	0.6578	0.6918
NetScience	0.8287	0.8243	0.8021	0.8318	0.8144	**0.9400**	0.9240	0.8435
Email	0.4326	0.4639	0.3502	0.4606	0.0000	**0.5400**	0.2990	0.5284
PowerGrid	0.7497	0.7423	0.7073	0.7647	0.5046	0.8200	0.8190	**0.9159**
PGP	0.6994	0.7146	0.6601	0.7578	0.2394	0.8200	0.8190	**0.8629**
GrQc	0.7503	0.7105	0.7063	0.7462	0.4361	0.7900	0.7940	**0.8277**
AstroPh	0.2931	0.2448	0.2529	0.3990	0.0773	0.5000	0.4528	**0.5629**
ca-HepTh	0.5727	0.5270	0.5348	0.5862	0.1730	0.7000	0.7060	**0.7331**
ca-HepPh	0.4425	0.4597	0.3823	0.5434	0.1440	0.4900	0.4726	**0.6138**
Condmat-2003	0.4898	0.4911	0.4610	0.5696	0.1010	**0.7000**	0.6640	0.6776
Condmat-2005	0.4337	0.4877	0.4069	0.4069	0.0763	**0.6600**	0.6320	0.6311
Email Enron	0.3310	0.2887	0.3030	0.3478	0.0786	0.5500	**0.5750**	0.5244
Collaboration	0.6838	0.6531	0.6427	0.7036	0.2306	0.7900	0.6251	**0.7967**
Internet	**0.5283**	0.5028	0.5075	0.3983	0.0050	0.4900	0.4128	0.4981

TABLE 8. The average rank of various algorithms in real-world benchmark networks.

Test Statistics	
N	18
Chi-Square	48.751
df	7
Asymp. Sig.	0.001

Mean Rank	
CMA	4.83
GAOMA-net	4.47
GACD	3.53
GATB	4.72
LabelRank	1.56
LBLD	5.50
CSLPR	4.81
LCD-SN	6.58

it could be mentioned that the average ranks in real-world networks had statistically significant differences. According to the average ranks, the LCD-SN algorithm won first, and the LBLD and CMA algorithms won second and third, respectively. In summary, it can be said that the LCD-SN algorithm is effective for discovering community structure in real-world benchmark networks.

E. EVALUATION WITH SYNTHETIC BENCHMARK NETWORK

In this section, the efficiency of the introduced algorithm is evaluated in LFR synthetic benchmark networks and is compared with CMA [14], GAOMA-net [13], GACD [11], GATB [12], LabelRank [34], LBLD [29], and CSLPR [17] algorithms. LFR synthetic benchmark networks are presented in [49] and are more suitable with the characteristics of

TABLE 9. NMI value obtained by various algorithms in LFR synthetic benchmark network.

μ	CMA	GAOMA -net	GACD	GATB	Label Rank	LBLD	CSLPR	LCD-SN
0.00	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
0.05	0.9988	0.9994	0.9267	**1.0000**	**1.0000**	**1.0000**	0.9846	0.9902
0.10	0.9604	0.9814	0.7335	0.9955	0.9997	**1.0000**	0.9708	0.9831
0.15	0.8568	0.9811	0.6293	0.9689	0.9576	**1.0000**	0.9537	0.9652
0.20	0.7271	0.9299	0.5134	0.9317	0.9010	0.9412	0.9382	**0.9520**
0.25	0.6197	0.7237	0.4207	0.8773	0.7337	0.9127	0.9125	**0.9278**
0.30	0.4928	0.6333	0.3203	0.8150	0.6493	0.8626	0.8847	**0.9044**
0.35	0.4084	0.4758	0.2352	0.6014	0.5063	0.8117	0.8132	**0.8352**
0.40	0.3076	0.3819	0.1966	0.3251	0.4835	0.7159	0.7071	**0.7494**
0.45	0.2369	0.1977	0.1674	0.2187	0.2000	0.4437	0.5921	**0.6147**
0.50	0.1717	0.1321	0.1205	0.1528	0.0927	0.4105	0.4414	**0.4678**

real-world networks. In LFR networks, node degrees distribution and community sizes distribution follow the power law distribution with exponents τ_1 and τ_2, respectively. The robustness of the community structure is related to the mixing parameter μ. The mixing parameter represents each node's average connections to other communities. The benchmark networks with a smaller mixing parameter have a significant community structure. The values used for different parameters of LFR networks are given below. In the experiments, the network size is set to 500, the maximum and average degree of nodes are set to 50 and 25, respectively, the values of τ_1 and τ_2 are set to 2 and 1, respectively, and the maximum and minimum size of communities are set to 100 and 50, respectively. The mixing parameter μ is considered from 0.00 to 0.50. Considering that the ground-truth community structure is known in LFR synthetic benchmark networks, the NMI criterion is utilized to evaluate the performance of different algorithms. The average NMI obtained from different algorithms is given in Table 9. In the non-deterministic algorithms (CMA, GAOMA-net, GACD, and GATB), each data is the average of 10 runs. The largest NMI value obtained for each network is written in bold. The LCD-SN has obtained the highest NMI value in all networks except for $\mu = \{0.05, 0.10, 0.15\}$. In these networks, LBLD has obtained the highest NMI value. For further comparison, the results of Table 9 are given as a plot in Figure 5. As it is observed, by increasing the value of the mixing parameter μ, since the network's community structure becomes ambiguous, all algorithms' efficiency decreases. This reduction in the LCD-SN, CSLPR, and LBLD is far less than that of other algorithms.

In addition, Table 10 shows the results of the Friedman test for ranking different algorithms in the LFR synthetic networks. As you can see, the Friedman test statistic value (p-value < 0.001) is less than 0.05. Therefore, with a probability of 0.95, it could be mentioned that the average ranks in LFR synthetic networks had statistically significant differences. According to the average ranks, the LCD-SN algorithm won first, and the LBLD and CSLPR algorithms won second and third, respectively. In summary, it can be said that the LCD-SN algorithm is effective for discovering community structure in LFR synthetic benchmark networks.

TABLE 10. The average rank of various algorithms in LFR synthetic benchmark networks.

Test Statistics	
N	11
Chi-Square	45.450
df	7
Asymp. Sig.	<0.001

Mean Rank	
CMA	2.95
GAOMA-net	3.86
GACD	1.41
GATB	4.95
LabelRank	4.23
LBLD	6.59
CSLPR	5.32
LCD-SN	6.68

F. RESOLUTION LIMIT

In [50], Lancichinetti et al. have shown that methods based on modularity optimization may fail to determine communities smaller than a specified scale, which is related to the total network size and interconnectedness of the communities. This problem is known as resolution limitation. In this section, the resolution limit of the introduced method is investigated. Consider the network in Figure 6. This network comprises 54 nodes and two communities, C_1 and C_2, with 50 and 4 nodes, respectively. A node in any community is connected to all other nodes of the same community. In other words, each community is a fully connected graph. The only edge connecting two communities is between node u in the community C_1 and node v in community C_2. The modularity value of the community structure in the network is 0.0097. However, modularity optimization-based approaches assign node u to the community C_2 because this network partition has a larger value of 0.0101 for modularity. This phenomenon is exactly because of the resolution limitation of modularity optimization.

LCD-SN algorithm, along with CMA, GAOMA-net, GATB, GACD, LabelRank, CSLPR, and LBLD algorithms, have been executed 100 times in the test network of Figure 6,

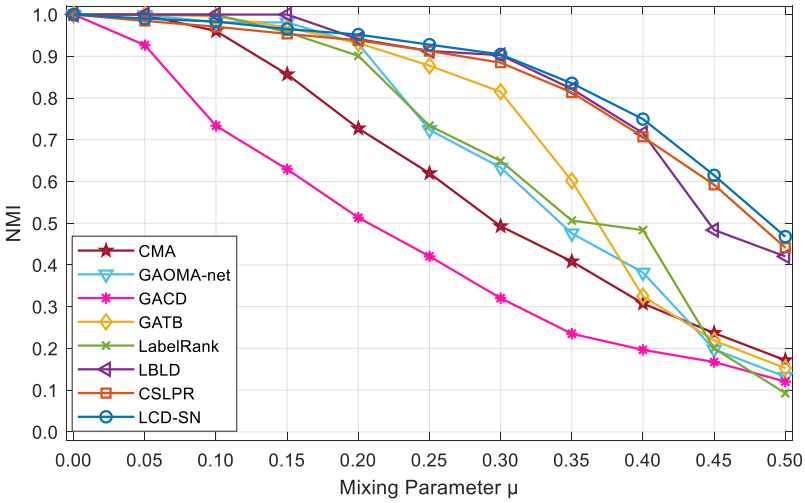

FIGURE 5. NMI value obtained using various algorithms in LFR synthetic benchmark network.

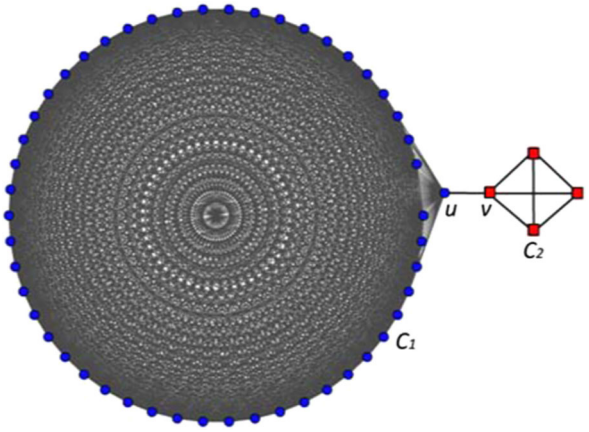

FIGURE 6. Sample network with 54 nodes to investigate the resolution limit.

and the results are reported in Table 11. Since CMA, GAOMA-net, GACD, and GATB algorithms are based on modularity optimization, in most cases, they wrongly assign node u as a member of the community C_2. It should be noted that the LabelRank algorithm identifies the entire network as a community in all runs. LCD-SN algorithm has correctly identified communities C_1 and C_2 in all runs. Therefore, our algorithm effectively solves the problem of resolution limitation.

G. DISCUSSION

This subsection explains the reasons behind the introduced algorithm's performance compared to other algorithms. As mentioned, the proposed algorithm first ranks the network nodes. In determining the rank of a node, its second-degree neighbors are also considered, in addition to its first-degree neighbors. Then, starting from the higher-rank nodes, initial local communities consisting of the corresponding node and its first-degree neighbors are formed. This is what happens in community formation in real-world networks. In real-world networks, different nodes usually form a community around important node. In short, the introduced algorithm identifies meaningful communities by fully using the network's structural information.

TABLE 11. The resolution limit of different algorithms.

Algorithm	The percentage of times that communities C1 and C2 are correctly identified
CMA	53%
GAOMA-net	1%
GACD	22%
GATB	0%
LabelRank	0%
LBLD	46%
CSLPR	31%
LCD-SN	100%

V. CONCLUSION AND FUTURE WORKS

This article presents a new method called LCD-SN to detect communities in social networks. The proposed method is a local method based on node ranking. LCD-SN algorithm consists of three phases. In the first phase, network nodes are ranked using a new criterion called IMP, and initial local communities are formed around high-ranked nodes. In the second phase, overlapping nodes are assigned to a single community using the GLHN similarity measure. In the third phase, the obtained communities in the previous phase are improved by post-processing (removing small communities and merging weak communities with strong ones). Among the advantages of the proposed algorithm its dependence on the minimum number of input parameters, its locality and no need for the information of the entire network, not having

the problem of resolution limit, and most importantly, its certainty could be mentioned. In this article, single-layer and unsigned networks are considered. The generalization of the proposed algorithm to identify the community structure in multi-layer, signed, and weighted networks could be the focus of our next work.

REFERENCES

[1] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Phys. Rep.*, vol. 533, no. 4, pp. 95–142, Dec. 2013.
[2] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.
[3] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.
[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.
[5] S. T. Shishavan and F. S. Gharehchopogh, "An improved cuckoo search optimization algorithm with genetic algorithm for community detection in complex networks," *Multimedia Tools Appl.*, vol. 81, no. 18, pp. 25205–25231, Jul. 2022.
[6] F. S. Gharehchopogh, "An improved Harris hawks optimization algorithm with multi-strategy for community detection in social network," *J. Bionic Eng.*, vol. 20, no. 3, pp. 1175–1197, May 2023.
[7] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, Feb. 1970.
[8] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.
[9] J. Xi, W. Zhan, and Z. Wang, "Hierarchical community detection algorithm based on node similarity," *Int. J. Database Theory Appl.*, vol. 9, no. 6, pp. 209–218, Jun. 2016.
[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mechanics, Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.
[11] C. Shi, Y. Wang, B. Wu, and C. Zhong, "A new genetic algorithm for community detection," in *Proc. 1st Int. Conf.*, 2009, pp. 1298–1309.
[12] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," 2007, *arXiv:0711.0491*.
[13] B. Zarei and M. R. Meybodi, "Detecting community structure in complex networks using genetic algorithm based on object migrating automata," *Comput. Intell.*, vol. 36, no. 2, pp. 824–860, May 2020.
[14] B. Zarei, M. R. Meybodi, and B. Masoumi, "Chaotic memetic algorithm and its application for detecting community structure in complex networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 1, pp. 013125-1–013125-17, Jan. 2020.
[15] B. Zarei, M. R. Meybodi, and B. Masoumi, "A new evolutionary model based on cellular learning automata and chaos theory," *New Gener. Comput.*, vol. 40, no. 1, pp. 285–310, Apr. 2022.
[16] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.
[17] S. Aghaalizadeh, S. T. Afshord, A. Bouyer, and B. Anari, "Improving the stability of label propagation algorithm by propagating from low-significance nodes for community detection in social networks," *Computing*, vol. 104, no. 1, pp. 21–42, Jan. 2022.
[18] B. Zarei, M. R. Meybodi, and B. Masoumi, "Detecting community structure in signed and unsigned social networks by using weighted label propagation," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 10, pp. 103118-1–103118-17, Oct. 2020.
[19] A. Torkaman, K. Badie, A. Salajegheh, M. H. Bokaei, and S. F. F. Ardestani, "A four-stage algorithm for community detection based on label propagation and game theory in social networks," *AI*, vol. 4, no. 1, pp. 255–269, Feb. 2023.
[20] S. Aghaalizadeh, S. T. Afshord, A. Bouyer, and B. Anari, "A three-stage algorithm for local community detection based on the high node importance ranking in social networks," *Phys. A, Stat. Mech. Appl.*, vol. 563, Feb. 2021, Art. no. 125420.
[21] X. Ding, J. Zhang, and J. Yang, "A robust two-stage algorithm for local community detection," *Knowl.-Based Syst.*, vol. 152, pp. 188–199, Jul. 2018.
[22] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104822.
[23] S. Wang, J. Yang, X. Ding, J. Zhang, and M. Zhao, "A local community detection algorithm based on potential community exploration," *Frontiers Phys.*, vol. 11, Feb. 2023, Art. no. 1114296.
[24] J. Cai, J. Hao, H. Yang, Y. Yang, X. Zhao, Y. Xun, and D. Zhang, "A new community detection method for simplified networks by combining structure and attribute information," *Expert Syst. Appl.*, vol. 246, Jul. 2024, Art. no. 123103.
[25] H. Zheng, H. Zhao, and G. Ahmadi, "Towards improving community detection in complex networks using influential nodes," *J. Complex Netw.*, vol. 12, no. 1, Dec. 2023, Art. no. cnae001.
[26] K. Berahmand, Y. Li, and Y. Xu, "A deep semi-supervised community detection based on point-wise mutual information," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 3, pp. 3444–3456, Jun. 2024.
[27] K. Berahmand, Y. Li, and Y. Xu, "DAC-HPP: Deep attributed clustering with high-order proximity preserve," *Neural Comput. Appl.*, vol. 35, no. 34, pp. 24493–24511, Dec. 2023.
[28] K. Berahmand, M. Mohammadi, R. Sheikhpour, Y. Li, and Y. Xu, "WSNMF: Weighted symmetric nonnegative matrix factorization for attributed graph clustering," *Neurocomputing*, vol. 566, Jan. 2024, Art. no. 127041.
[29] H. Roghani and A. Bouyer, "A fast local balanced label diffusion algorithm for community detection in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5472–5484, Jun. 2023.
[30] S. Brin, "The PageRank citation ranking: Bringing order to the web," in *Proc. ASIS*, vol. 98, 1998, pp. 161–172.
[31] L. C. Freeman, "Centrality in social networks: Conceptual clarification," in *Social Network: Critical Concepts in Sociology*, vol. 1. Evanston, IL, USA: Routledge, 2002, pp. 238–263.
[32] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 73, no. 2, Feb. 2006, Art. no. 026120.
[33] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiology*, vol. 54, no. 4, pp. 396–405, Sep. 2003.
[34] J. Xie and B. K. Szymanski, "LabelRank: A stabilized label propagation algorithm for community detection in networks," in *Proc. IEEE 2nd Netw. Sci. Workshop (NSW)*, Apr. 2013, pp. 138–143.
[35] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 9, Sep. 2005, Art. no. P09008.
[36] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.
[37] V. Krebs, "The political books network," unpublished, doi: 10.2307/40124305.
[38] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI Conf. Artif. Intell.*, 2015, no. 1, pp. 1–12.
[39] R. Guimer, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 6, Dec. 2003, Art. no. 065103.
[40] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, nos. 66–84, pp. 440–442, 1998.
[41] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 056122.
[42] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 2, Mar. 2007.
[43] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2005, pp. 177–187.

[44] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066133.
[45] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.
[46] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, Jan. 2009.
[47] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, Aug. 2005, Art. no. 027104.
[48] B. Karrer, M. E. J. Newman, and L. Zdeborova, "Percolation on sparse networks," *Phys. Rev. Lett.*, vol. 113, 2014, Art. no. 208702. [Online]. Available: http://arxiv.org/abs/1405.0483
[49] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.
[50] A. Lancichinetti and S. Fortunato, "Limits of modularity maximization in community detection," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 6, Dec. 2011, Art. no. 066122.

JAFAR SHEYKHZADEH was born in Iran, in 1977. He received the B.S. degree in computer engineering from Islamic Azad University, Lahijan Branch, Lahijan, Iran, in 2001, and the M.S. degree in computer engineering from Islamic Azad University, Qazvin Branch, Qazvin, Iran, in 2009. He is currently pursuing the Ph.D. degree in computer science with Islamic Azad University, Urmia Branch, Urmia, Iran. He has been an Instructor with the Department of Computer Engineering, Islamic Azad University, Ahar Branch, Ahar, Iran, since 2009. His research interests include task scheduling, complex, and social networks analysis.

BAGHER ZAREI received the B.Sc. degree in computer engineering from Islamic Azad University, Shabestar Branch, Shabestar, Iran, in 2004, and the M.Sc. and Ph.D. degrees in computer engineering from Islamic Azad University, Qazvin Branch, Qazvin, Iran, in 2006 and 2020, respectively. He is currently an Assistant Professor with the Department of Computer Engineering and Information Technology, Islamic Azad University, Shabestar Branch. His research interests include software development, soft computing, complex network analysis, reinforcement learning, and chaos theory.

FARHAD SOLEIMANIAN GHAREHCHOPOGH was born in Iran, in 1979. He received the B.S. degree in computer engineering from Islamic Azad University, Shabestar Branch, Iran, in 2002, the M.S. degree in computer engineering from Cukurova University, Adana, Türkiye, in 2011, and the Ph.D. degree in computer engineering from Hacettepe University, Ankara, Türkiye, in 2015. He has been an Assistant Professor of computer engineering with Islamic Azad University, Urmia Branch, Urmia, Iran, since 2015. He has published more than 150 papers in international journals and conferences. His research interest includes metaheuristic algorithms and their application issues.