IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 2, 2019, accepted December 15, 2019, date of publication December 18, 2019,
date of current version December 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960574

Parallel Heuristic Community Detection Method Based on Node Similarity

QIANG ZHOU[1], SHI-MIN CAI[1,2], AND YI-CHENG ZHANG[2,3]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Institute of Fundamental and Frontier Science, University of Electronic Science and Technology of China, Chengdu 611731, China
[3]Department of Physics, University of Fribourg, 1700 Fribourg, Switzerland

Corresponding author: Shi-Min Cai (shimin.cai81@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673086, and in part by the Science Promotion Programme of the University of Electronic Science and Technology of China (UESTC), China, under Grant Y03111023901014006.

ABSTRACT Community structure discovery can help us better understand the capabilities and functions of the network. However, many existing methods have failed to identify nodes in communities accurately. In this paper, we proposed a heuristic community detection method based on node similarities that are computed by assigning different edge weight influence factors based on different neighbor types of nodes. Concretely, by arbitrarily choosing a pair of nodes, we firstly found out the common neighbor nodes of the node pair and their corresponding neighbor nodes. Then, different edge weight influence factors are assigned according to the impact of different types of neighbor nodes on node similarity. Finally, the similarities between a pair of nodes are calculated by the proportion of various edge weight influence factors related to the node pair. Along the direction, a hash table based data storage and retrieval strategy with a lower conflict rate is introduced to hash the edge information into a ternary bucket structure that can be merged according to the same starting node. This operation can reduce the time complexity of the data query to a constant level, and realize the parallel computing of node similarity. When obtaining similarity of node pair, we merged nodes into communities by a heuristic hierarchical clustering. And, the resulting community structure is detected until all node similarities are calculated. With the help of the comparison tests of different methods based on the benchmark networks that have ground-truth communities, the proposed method for community detection provides better performance in both identification accuracy and time efficiency.

INDEX TERMS Community detection, complex network, node similarity, hash table, parallel heuristic strategy, hierarchical clustering.

I. INTRODUCTION

Nature is a complex system of mutual interaction and polymorphism. Its commonality behaves relatively complicated internal structure that can be mapped into a nonlinear data structure similar to a graph (or network). In a network, nodes represent individuals and the edges indicate the tie among individuals. Community, one of the structure unit in the network, has play an important role on understanding the network capabilities [1]–[3]. For a network with community structure, similar nodes closely linked with more edges are classified into diverse communities according to the topological and attribute characteristics [4]–[10]. The community structure is common in a variety of complex networked systems, and the functions and roles among diverse communities are also not the same. For example, it may describe individual groups with common interests or a set of common topics in a social network [11] and a group of common living habits in a biosphere [12], as well as control the disease outbreak in a epidemic spreading [13]. As one of the hot topics in network science and computer science fields, community detection is worth to be efficiently investigated.

Global-based community detection in large-scale complex networks is a NP hard problem [14]. However, approximately heuristic methods can be used for community detection within a reasonable time efficiency. Most of them treat network as a one-dimension model. As the studies progress, the researchers found that in real world there are still bipartite networks constructed with two different types of

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

nodes, in which the community detection is a little different [15]–[17]. Although the edges in bipartite networks only exist between heterogeneous nodes, we can project bipartite network into a one-dimension model by connecting same type of nodes with their common neighbors. Thus, we herein mainly focus on the one-dimension model of network and explore the corresponding community structure accurately.

Although many existing community detection methods can achieve relatively high modularity due to the lack of consideration of inter-node correlation, there is still the problem of node misclassification. Considering this point, this paper proposed a more effective secondary decision rule for evaluating node similarity, which is equivalent to add certain extra attributes to the node. So that, when faced with a overlapping node, it can use the redivided network topology to achieve higher community division accuracy in comparison of the Jaccard similarity [18]. The proposed rule can be extended to weighted network with overlapping community structure. Thus, based such measurement of node similarity, the parallel heuristic community detection method proposed in this paper can be applied to one-dimension model of network in most cases.

In the next, we simply illustrate the parallel heuristic community detection method based on node similarity. As mentioned-above, the measurement of node similarity is a key strategy for accurately detecting community structure. The secondary decision rule both considers the local topological information and internal correlations of nodes to evaluate node similarity. In the process of evaluating node similarity, we firstly find out the common neighbor nodes of a pair of the node pair, and by bridging these common neighbor nodes, further explore their corresponding neighbor nodes. Then, we assign different edge weight influence factors according to the impact of different types of neighbor nodes on node similarity. Finally, we calculate the similarity between such pair of nodes by the proportion of various edge weight factors related to the node pair. Based on the similarities between pairs of nodes, we use a heuristic hierarchical clustering to merge nodes into communities. Note that as the communities are heuristically obtained in an agglomerated way, the results of community detection isn't related to the selection of initial node.

In addition, when calculating the similarities of all pairs of nodes, a parallel approach is obviously more efficient [18]–[21]. For that, a hash table based data storage and retrieval strategy is proposed by developing a dynamic node storage hash table. It is based on two distinct hash tables, one stores the edge information of endpoint node, the other stores the edge information of another endpoint node. With such strategy, it is not necessary to reestablish a new edge mapping relationship after each calculation of node similarity, and a dynamic management is implemented in order to achieve the parallel table look-up calculation of node similarity with little overhead. Furthermore, the Fibonacci hashing function [22], [23] is introduced to balance the contradiction between the storage and occupied hash buckets of the node data structure, which greatly saves computer resources.

Based on the above-mentioned analysis, we summarized the main contributions of this work:

1) We proposed the secondary decision rule for better evaluate node similarity. The novel node similarity criteria both considers the local topological information and internal correlations of nodes.
2) In the process of evaluating node similarity, we proposed a hash table based data storage and retrieval strategy with a lower conflict rate. It realizes the parallel computing of node similarity to greatly improve the computational efficiency.
3) Combining with (1) and (2), we proposed a parallel heuristic community detection method to discovery the network community structure more accurately.

The rest of the paper is organized as follows: firstly, the work related to our study is introduced in section II and the related research strategies about the proposed method are illustrated in section III, including the node similarity criteria, the hash table based parallel computing of node similarity and the description of algorithm principal frame; then the section IV shows the experimental results of the proposed method, including the metrics, material and algorithm evaluation; and the section V presents detailed discussions on the experimental results from detection accuracy and computational efficiency; Finally, in section VI, we concluded our work.

II. RELATED WORK

The ideal situation of community detection is that taking into account both the network topological structure and node attribute characteristics. The network topological structure commonly determines the global properties of communities, and the node attribute characteristics are more important for local fine-tuning of communities [24]. For example, when determining a intersection node potentially belonging to overlapping communities, its own attribute characteristics often has a definite effect [25]. To fully consider the global and local properties of the network in community detection, many solutions have been proposed. A scalar objective function based on modularity is widely used to determine the number of existing communities and the division quality of community detection [26]. The higher the score of modularity, the more it can truly reflect the community division. Although the modularity-based community detection methods are widely suggested, they have a resolution problem that the communities with relatively small number of nodes are hard to be detected [27], [28]. Wang et al. proposed a method of using core-vertex and intimate degree to detect the community [29]. It builds up the community structure of network by finding the core-vertex in the original network and then calculating the intimacy of the new members. Its advantage is that ordinary nodes in the network can be detected more precisely, but it takes extra time to find the existing core-vertex. Eustace et al. proposed a local community neighbor-

hood ratio function, which predicts the network community structure by detecting the neighbor nodes with overlapping phenomena [30]. Its advantage is that the similarity of local structures is used to infer the ownership of nodes, but does not take into account the interaction of indirect neighbors between nodes. Cui et al. designed a maximal sub-graphs and node belonging degrees to discovery the overlapping community structures [31]. The main idea of the algorithm is to find the key pair-vertices and then merge the maximum sub-graphs containing the key pair-vertices iteratively. It can find all the biggest sub-graphs and the overlapping nodes accurately in the network, but the properties of the node itself are not fully considered. Although the above-mentioned algorithms can better complete the task of community detection in the network, but still fail to fully consider the various neighbor relationships between nodes. Therefore, next we will illustrate the discovery strategy of parallel heuristic community based on node similarity proposed in this paper.

III. RELATED STRATEGIES OF PARALLEL HEURISTIC COMMUNITY DETECTION METHOD

A. NODE SIMILARITY CRITERIA

Herein, we mainly illustrate the secondary decision rule for the node similarity criteria. Supposing that there is a network G with N nodes and M edges, according to the incident relationship of its edges, the adjacency matrix of network can be constructed, as shown in the following formula,

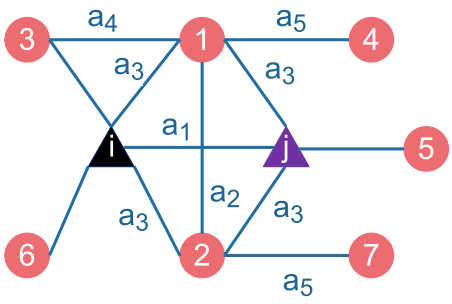
$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}, \tag{1}$$

where the cell $A_{ij} = 1$ indicates that there exists an edge from node i to j, otherwise $A_{ij} = 0$. So, we can abbreviate the adjacency matrix to the following formula,

$$A_{ij} = \begin{cases} 1, & \text{if node } i \text{ and } j \text{ are connected,} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

Through the adjacency matrix, the connections among nodes can be observed more intuitively. In network G, the degree of a node associates with the number of edges linked to that node, reflecting its local topological information. For a directed network, it is necessary to consider the out-degree and in-degree of nodes. Analogously, the degree of node i involving with community C can be interpreted as the number of edges of node i linked to the community C. Intuitively, the greater the degree, the closer the connection between node i and community C, so that node i is more likely to belong to community C. In addition, the concept of a neighbor node refers to another node in the network that has a direct connection with a certain node. Obviously, the degree of node equals to the number of its neighbor nodes. The set of neighbors for all nodes in a community C can be defined as follows,

$$neighbor_C = \bigcup_{i}^{N} neighbor_i, \tag{3}$$



$$sim(i,j) = \frac{a_1 * 1 + a_2 * 1 + a_3 * 4 + a_4 * 1 + a_5 * 2}{1 + 1 + 4 + 1 + 2} = \frac{1 * 1 + 0.8 * 1 + 0.6 * 4 + 0.2 * 1 + 0.1 * 2}{9} = 0.5111$$

FIGURE 1. The schematic illustration of the node similarity criteria based on secondary decision rule.

which can be used as a basic rule for defining node similarity later.

The mutual connections among nodes play a key role in community detection, and the more commonly used one is the node similarity determination. Determining node similarity is mainly based on the topological structure of network. For more precisely evaluating node similarity, some other attributes (they don't necessarily refer to a specific form or interpretation) of nodes may also have to be considered. A large number of evaluation methods of node similarity have been proposed, such as the similarity matrix based distance [32], the Pearson correlation between columns or rows of the adjacency matrix [33], the Jaccard similarity that considers the number of common neighbor nodes [4], and the random walk based on the measurements of node similarity [34]. However, most of these methods only emphasize the common neighbor relationships among nodes, but ignore the relationships among the secondary neighbor relationships among the common neighbor nodes and their corresponding neighbor nodes. In order to overcome the non-trivial flaw, we proposed the secondary decision rule to use more extra attributes of nodes for evaluating node similarity accurately.

The node similarity criteria based on the secondary decision rule mainly involves with tow aspects, the first one associates with the secondary neighbor relationships among the common neighbor nodes and their corresponding neighbor nodes, and the second one introduces the edge weight influence factor for the diverse secondary neighbor relationships. The key idea is summarized in Fig.1. For a pair of nodes i and j, we find their common neighbor nodes (e.g., node 1 and 2) and their corresponding neighbor nodes (e.g, nodes 3 and 4 of node 1, node 7 of node 2). In such local topological structure, we can determine the five types of connecting relationships, and assign different edge weight influence factors with $a_1 > a_2 > a_3 > a_4 > a_5$,

1) For directly connected edges between nodes i and j, we assign it with a_1.

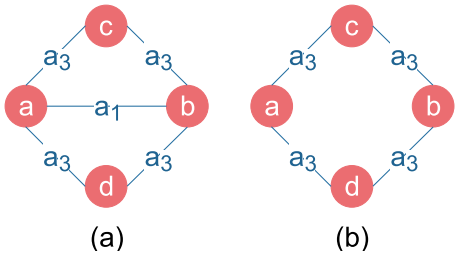


FIGURE 2. An example of node similarity comparison using the proposed method and Jaccard similarity. For a pair of nodes *a* and *b*, according to Eq.(4) and Eq.(5), (a) the nodes similarity is respectively 0.68 and 0.5; (b) the nodes similarity is respectively 0.6 and 1.

2) For common neighbor between nodes i and j, if there is directly connected edge between them, we assign it with a_2.
3) For common neighbor between nodes i and j, if there is directly connected edge with node i or j, we assign it with a_3.
4) For common neighbor between nodes i and j, if there is indirectly connected edge with node i or j, we assign it with a_4
5) For other case, we assign it with a_5.

Herein, we consider that the different impacts of edge weight influence factors on node similarity, so that the node similarity criteria is determined by the following equation,

$$sim(i,j) = \frac{a_1 N_{a_1} + a_2 N_{a_2} + a_3 N_{a_3} + a_4 N_{a_4} + a_5 N_{a_5}}{N_{a_1} + N_{a_2} + N_{a_3} + N_{a_4} + N_{a_5}}, \quad (4)$$

where N_{a_i} represents the number of the five kinds of connection. As shown in Fig.1, we presented an example that the similarity between nodes i and j is calculated to be 0.5111 in restricted to the specific values of a_i (that is $a_1 = 1, a_2 = 0.8$, $a_3 = 0.6, a_4 = 0.2$, and $a_5 = 0.1$). In the current work, these values of a_i are same in the following experiments.

Then, we also introduce the Jaccard similarity as the contrast evaluation of node similarity, defined as follows,

$$Jaccard = \frac{neighbor(i) \cap neighbor(j)}{neighbor(i) \cup neighbor(j)}, \quad (5)$$

where $neighbor(i)$ and $neighbor(j)$ are the neighbor collections of nodes i and j respectively. As can be seen from Eq.(5), due to computing the intersection and union of nodes, it brings a lot of computation time to realize the merge and separate operations.

We have presented an example to illustrate the accuracy of the proposed method in comparison of the Jaccard similarity. Fig.2 shows two types of local topological structures for a pair of node a and b. According to Eq.(4) and Eq.(5), we respectively obtain the node similarity with 0.68 and 0.5 in Fig.2a, and analogously 0.6 and 1 in Fig.2b. Thus, we can see that in Fig.2a, the proposed method is superior to the Jaccard similarity in terms of accuracy (i.e, $0.68 > 0.5$), while in Fig.2b, the Jaccard similarity seemly show the more precise similarity because it only considers the symmetry of the graph, but neglects the impact of different types of neighbor nodes. Nevertheless, for a pair of nodes a and b, the local topological information in Fig.2a is obviously richer than that in Fig.2b. Naturally, the node similarity in Fig.2a should be larger than that in Fig.2b. The proposed method exactly behaves in line with the expectation because it evaluates the node similarity (0.68) in Fig.2a larger than that (0.6) in Fig.2b, however, the Jaccard similarity is opposite to the expectation. Through the above-mentioned analysis, we can see that the proposed method is closer to a rational judgment because it considers more local topological information.

B. PARALLEL COMPUTING STRATEGY

In the face of growing online network data, real networked systems become more complicated with a large-scale topological structure. The traditional serial computing strategy is obviously unable to respond quickly and efficiently because of waiting for the calculation of resource consumption and waste. It strongly affects the computing efficiency although the serial computing strategy has certain advantages over parallel computing strategy from the perspective of reliability and security and the parallel computing strategy is hard to be designed [18], [19]. Thus, herein, we urge us to propose a parallel computing strategy for evaluating node similarity to efficiently utilize the computer resources.

In the process of evaluating node similarity, how to quickly retrieve the edge (weight) information of neighbor nodes is a key step, which directly determine the efficiency of community detection method. The main challenge is how to store and maintain a table of edge (weights) information that will change over time. Herein, we investigate a software approach by a hash table based data storage and retrieval strategy. A hash table is such a data storage model where large-scale structure data can quickly realize the operations of query, insert and delete in a near constant time level [35]. Meanwhile, considering that the Fibonacci hash function is more uniform in the spatial structure of data allocation, and its hashing conflict is minimal, we thus employ it to parallel computing strategy [22].

When designing the hash table abased data storage and retrieval strategy, we use two hash tables to represent the node information in a directed graph, one stores the information of the incident edge of the node, and another one stores the information at the launch side. Then, according to calculation requirement, the one-dimension decomposition model is used to classify the nodes and their edge list linearly. Each node is assigned to its corresponding set according to the Fibonacci hash function. The same node is responsible for information management of all nodes, edges and weights associated with it. In Fig.3, we simply illustrate the basic framework for this storage strategy. Concretely, the nodes are firstly classified according to their incident direction. The table-in deals with the incoming edges, and the table-out deals with the outgoing edges. $key(x)$ is an objective function of a tuple $g(i,j)$ which is defined in Eq.(6),

$$g(i,j) = j|(i \ll 16), \quad (6)$$

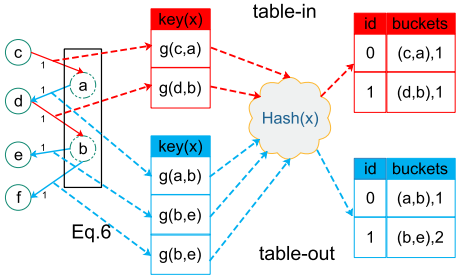


FIGURE 3. A hash table based data storage and retrieval strategy for storing node information.

where | is a bitwise OR operator and $\ll$ is a bitwise left shift operator. For the table-in, its tuple includes the source node i and the destination node j, while for the table-out, its tuple includes the source node j and the destination node k. Note that in Fig.3, the establishment process of $g(b, f)$ considers that the same starting node is responsible for managing its associated nodes, and hash tables are hashed on edges. Combining Eq.(6), it is not difficult to see that it has the same attributes as $g(b, e)$, so it is represented as the same node tuple $g(b, e)$. And, this is also to facilitate the merging of elements in the hash bucket.

In addition, when calculating the Fibonacci hash function, the result is stored in a ternary group $((i, j), \omega_{i,j})$. For a same group, the weight value is combined because of all these related edges are hashed into the same bucket in the table-out (see in Fig.3). Herein, the Fibonacci hash function is used in the experiment to make the distribution of the hash list more homogeneous and prevent the large-scale hash conflict [36]. It is defined as follows,

$$H(x) = \left\lfloor \frac{M}{W} \cdot ((\phi^{-1} \cdot W \cdot x) \bmod W) \right\rfloor. \tag{7}$$

where M is the size of the hash table, W is equal to $2^{64} - 1$, and ϕ is called *the golden ratio* [37].

To sum up, in the process of evaluating node similarity, the data storage involving with the local topological information of the nodes is constructed as a hash table. Based on the hash table, it is possible to efficiently retrieve the tuple data of the neighbor nodes linked to the objective node. With this strategy, it is not necessary to scan entire topological structure of network when the similarity between each pair of nodes is calculated. Merging the data of the same tuple not only improves the efficiency of hash table searching in operation, but also makes it possible to maintain the neighbor relationships of nodes dynamically when the network topology changes. Thus, such parallel computing strategy greatly improves the computation cost and reduce the time complexity of evaluating node similarity. Moreover, the hash table based data storage and retrieve strategy provides a novel idea for solving similar problem and is transferred to applications.

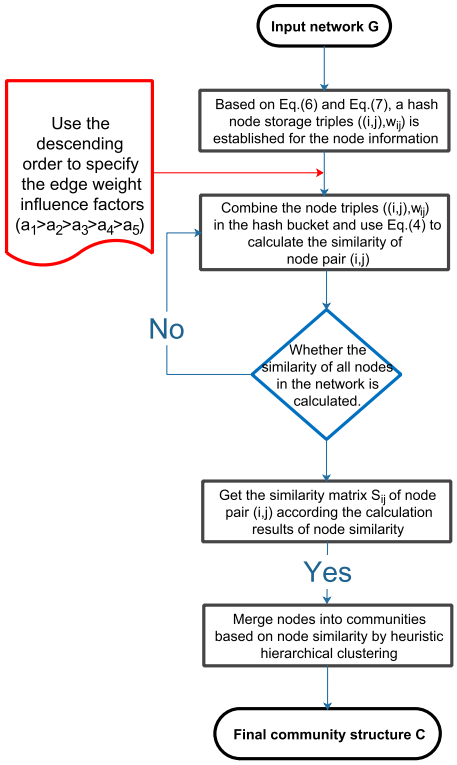


FIGURE 4. This flow chart of entire algorithm principal frame of parallel heuristic community detection method.

C. DESCRIPTION OF ALGORITHM PRINCIPAL FRAME

After deeply analyzing the node similarity criteria and its parallel computing strategy, we then construct the parallel heuristic community detection method. We have known that these nodes in a same community behave a similar attribute. Based on this concept, if the similarity of two nodes is large enough, they are highly possible to be divided into a same community. Based on the node similarity criteria and parallel computing strategy, we construct the parallel heuristic community detection method by the heuristic hierarchical clustering in an agglomerated way. Base on the constructed method, we can obtain the dendrogram of network to clearly discover the corresponding community structure. Thus, the flow chart of entire algorithm principal frame of proposed method can be described in the Fig.4, which is divided into three key parts.

The first part is that the hash table based data storage of node information according to Eq.(6) and Eq.(7). Its pseudocode is shown in Algorithm 1. Specifically: the first line shows that for the adjacent matrix A_{ij} in a given network G, the corresponding hash tuple storage table $g(i, j)$ is established according to the objective function in Eq.(6). For the table-in, its tuple includes the source node i and the destination node j in the second line, while for the table-out, its tuple includes the source node j and the destination node k in the third line. For the obtained table-in and table-out, the fourth line uses Fibonacci hash function in Eq.(7) to hash them and store the tripe group $((i, j), \omega_{i,j})$ in the hash buckets. Then, the fifth and sixth lines are merged according to the same

Algorithm 1 The Hash Table Based Data Storage of Node Information

Input: Adjacent matrix A_{ij} for a given network G
Output: The ternary group $((i, j), \omega_{i,j})$ of nodes
Procedure:
(1) $g(i, j) \leftarrow$ use $key(x)$ in Eq.(6), $\forall(i, j) \in A_{ij}$;
(2) table-in $\leftarrow g(i, j), \forall(i, j) \in A_{ij}$;
(3) table-out $\leftarrow g(j, k), \forall(j, k) \in A_{ij}$;
(4) $((i, j), \omega_{i,j}) \leftarrow$ use $Hash(x)$ in Eq.(7);
(5) for table-in: if $i = i$, $((i, j), \omega_{i,j}) \leftarrow$ $((i, j), \omega_{i,j} + \omega_{i,a}), \forall(i, a) \in A_{ij}$;
(6) for table-out: if $j = j$, $((j, k), \omega_{j,k}) \leftarrow$ $((j, k), \omega_{j,k} + \omega_{j,a}), \forall(j, a) \in A_{ij}$.

starting nodes in table-in and table-out respectively, and their weights are summed.

The second part is the node similarity evaluation of each pair of nodes according to Eq.(4). Its pseudo-code is shown in Algorithm 2. Specifically: the first line illustrates that according to different node neighbor types, different edge weight influence factors are assigned to node pairs (i, j) as shown in Fig.1. Then, query the node triple group $((i, j), \omega_{i,j})$ information stored in the hash bucket, and calculate the similarity between nodes using the node similarity criteria in Eq.(4), which is described in the second and third lines.

Algorithm 2 The Similarity Evaluation of Pairs of Nodes

Input: The ternary group $((i, j), \omega_{i,j})$ of nodes
Output: The node similarity matrix S_{ij}
Procedure:
(1) Assign $a_1 > a_2 > a_3 > a_4 > a_5$ to node pairs (i, j), $\forall(i, j) \in ((i, j), \omega_{i,j})$;
(2) **for** $(i, j) \in ((i, j), \omega_{i,j})$ **do**
(3) $S_{ij} \leftarrow$ use $sim(i, j)$ in Eq.(4);

The third part is that merging nodes into communities based on node similarity by heuristic hierarchical clustering in an agglomerated way. Its pseudo-code is shown in Algorithm 3. Specifically: the first line assumes that each node i is initially assigned to a separate community C_i, while the second to ninth lines illustrate a community heuristic hierarchical clustering process based on the node similarity criteria proposed in this paper. For the fifth and sixth lines, start with an arbitrary new node i and merge it into the community $C_{i,j}$ with node j (node j has the greatest similarity with node i), if node j is not included in the community where node i is. In addition, the seventh and eighth lines reflect that if node j is already included in the community where node i is, then jump to the second line to select nodes that have not been visited and continue the same hierarchical clustering process.

In addition, in order to illustrate the relevant parameters used in the paper more intuitively, we summarized the various symbols in Table 1.

IV. METHOD TESTING AND EVALUATION

In this section, we designed a series of performance tests on benchmark networks for evaluating the proposed community detection method. The benchmark networks are divided into two categories, one includes the real networks and the other includes the artificial networks based on LFR model [38]. In order to ensure the universality of the proposed community detection method, the topological structural characteristics such as the directionality and weight of the network are preserved in the calculation. The metrics on method testing mainly include the normalized mutual information, quality evaluation function and computational time, based on which the method evaluation is illustrated as follows.

Algorithm 3 The Community Detection Based Node Similarity by Heuristic Hierarchical Clustering

Input: The node similarity matrix S_{ij}
Output: Final community structure C in a given network G
Procedure:
(1) Assume $i \subseteq$ community $C_i, \forall i \in S_{ij}$;
(2) **While**($\forall i \in S_{ij} \bigcap (i$ is not visited))
(3) {
(4) $\forall j \in S_{ij}$;
(5) if(maximum($sim(i, j)$) $\bigcap (j \nsubseteq C_i)$)
(6) { $C_{i,j} \leftarrow merge(C_i, C_j)$; }
(7) else if($j \subseteq C_i$)
(8) {go to (2);}
(9) }

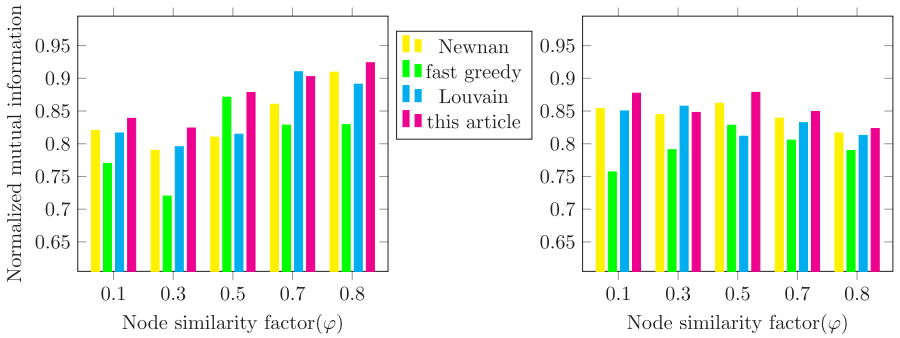
TABLE 1. Notation note for each symbol.

Symbol	Comments
G	The network
N	The number of nodes
M	The number of edges
A_{ij}	Adjacent matrix of node pair (i, j)
C	Community structure
a_i	The edge weight influence factor
N_{a_i}	The number of edge connections corresponding to a_i
table-in	Store the information of the incident edge of the node
table-out	Store the information of the outgoing edge of the node
$key(x)$	an objective function of a tuple $g(i, j)$
$((i, j), \omega_{i,j})$	The ternary group of nodes information
$\omega_{i,j}$	Edge weight information
S_{ij}	The node similarity matrix of node pair (i, j)
C_i	The community to which node i belongs
φ	The Jaccard node similarity factor
NMI	Normalized mutual information
Q	The quality (evaluation) function
μ	The mixing parameter of LFR network
β	The sizes of the communities of LFR network
γ	Node degree distribution of LFR network
$\langle k \rangle$	The average degree of LFR network

A. NORMALIZED MUTUAL INFORMATION

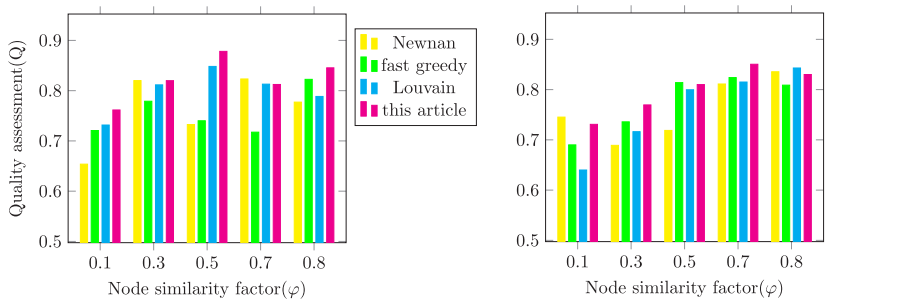
Normalized mutual information (NMI) [7] is a common way to determine the quality of community detection based on the ground truth of network with community structure. The metric is defined by calculating the NMI between the resulting community structure P^u and the corresponding ground truth P^v,

$$I(P^u, P^v) = \frac{X}{Y}, \tag{8}$$



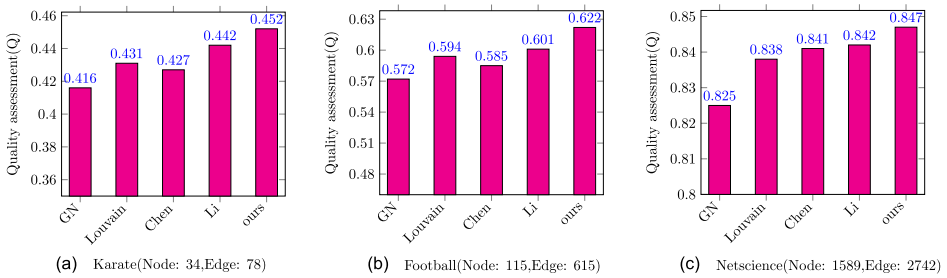
(a) NMI based on the email-EuAll dataset (b) NMI based on the com-Amazon dataset

FIGURE 5. Comparative analysis of NMI in restricted to different influence factor φ of Jaccard similarity.



(a) Quality assessment based on the email-EuAll dataset (b) Quality assessment based on the com-Amazon dataset

FIGURE 6. Comparison analysis of quality assessment Q in restricted to different influence factor φ of Jaccard similarity.



(a) Karate(Node: 34,Edge: 78) (b) Football(Node: 115,Edge: 615) (c) Netscience(Node: 1589,Edge: 2742)

FIGURE 7. Comparative analysis of quality assessment Q for diverse community detection methods in three real benchmark networks.

where

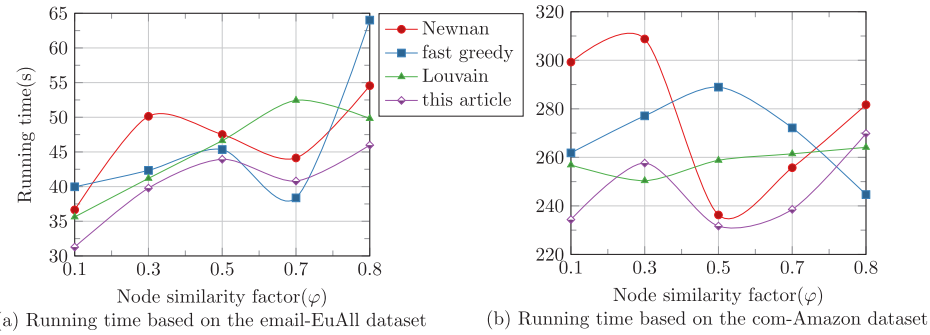
$$X = \sum_{i=1}^{c^u}\sum_{j=1}^{c^v} n_{i,j}\log\left(\frac{n\cdot n_{i,j}}{n_i^u\cdot n_j^v}\right), \tag{9}$$

and

$$Y = \sqrt{\left(\sum_{i=1}^{c^u} n_i^u \log\left(\frac{n_i^u}{n}\right)\right)\left(\sum_{j=1}^{c^v} n_j^v \log\left(\frac{n_j^v}{n}\right)\right)}. \tag{10}$$

In the above equations, c^u and c^v are the number of communities in P^u and P^v respectively. $n_{i,j}$ is the number of nodes assigned to i^{th} community in P^u, and j^{th} community in P^v. The number of nodes in P^u assigned to i^{th} community can be replaced by n_i^u, so the number of nodes in P^v assigned to j^{th} community can be also replaced by n_j^v. n represents the total number of nodes in the division. If P^u and P^v are uncorrelated, which tells you nothing but the mutual information is zero. Otherwise, the value determines how similar the two communities are.

The experiments use two real benchmark networks with ground truth, the email-EuAll [39] and com-Amazon [40] (mainly used in Fig.5, Fig.6 and Fig.8). The email-EuAll network is about all incoming and outgoing email between members in a research institution with 42 departments (or communities). The number of the nodes and edges are 1005 and 25571 respectively. The com-Amazon is a co-purchased network based on customers who bought this item also bought similar feature of the Amazon website. It contains 334863 nodes and 925872 edges.The objective facts reached 5000



(a) Running time based on the email-EuAll dataset (b) Running time based on the com-Amazon dataset

FIGURE 8. Comparison analysis of time performance in restricted to different influence factor φ of Jaccard similarity.

communities with the highest quality and group less than three are not considered as community. The average clustering coefficients of these two databases are 0.3994 and 0.3967, respectively.

In order to emphasize the importance of the central node in the network (generally, the greater the degree of the node, the more important the role it plays), the experiments herein do not directly use the original network for method testing. Instead, we firstly divide the original network into communities with Jaccard similarity according to Eq.(5). The different node similarity factors of Jaccard similarity are considered to obtain the new networks at the community scale. Then, we further perform the method testing based on these new community-scale networks. Such operation is greatly benefited to the detection of those overlapping (or margin) nodes that belong to multiple communities at the same time or that have a small degree, which can be used to determine whether they really belong to the community in which it is currently located. Moreover, the community structure is again in-depth excavation, allowing it to achieve community partition more accurately and obtain the better performance scores in comparison of the original network. We use three community detection method based on the greedy strategy to compare them with the parallel heuristic community detection method. The comparative results are obtained by averaging 100 times experiments.

Fig.5 shows the results of NMI in restricted to a series of influence factor φ of Jaccard similarity. The comparative methods based the greedy strategy are Newman fast algorithm (short of 'Newman') [41], the fast greedy algorithm (short of 'fast greedy') [42], and the Louvain algorithm (short of 'Louvain') [43] (mainly used in Fig.5, Fig.6, and Fig.8). As shown in Fig.5a, we can see that with increasing φ, the values of NMI obtained from the proposed method are generally higher than those from the comparative methods although the proposed method is much closer to fast greedy for $\varphi = 0.5$ and Louvian $\varphi = 0.7$. The potential reason is that when calculating the node similarity, the common neighbors of the nodes and their corresponding neighbor nodes are fully considered in the proposed method, but partially neglected in the others. We note that for $\varphi = 0.8$, the value of NMI is the highest, which is mainly brought by the redistribution of the edge weight influence factors. To a certain extent, it suggests that the intimacy (i.e, closer connections) among nodes has an important impact on the community partition. Analogously, the comparison analysis of NMI based on com-Amazon dataset is shown in Fig.5b. It can be seen that the values of NMI obtained from the proposed method is also generally higher than those from the comparative methods although the proposed method is much closer to Louvain for $\varphi = 0.3$. Nevertheless, the differences of NMI among these methods are not as big as those in email-EuAll dataset.

In addition, according to the node similarity criteria, the node similarity is calculated based on the local topological information of the network. To further illustrate its importance in community detection, we also compared it with other seven structure similarity indexes. Herein, we directly use diverse types of node similarity index to detect community structure of diverse networks. The comparative node similarity indexes are the Jaccard index (short of 'Jaccard') [7], the preferential attachment (short of 'preference') [44], the Sørensen index (short of 'Sørensen') [45], the TJA-net index (short of 'TJA-net') [46], the Leicht-Holme-Newman index (short of 'LHN') [47], the Salton index (short of 'Salton') [48] and the hub depressed index (short of 'HDI') [4]. We use two real benchmark networks, Zachary Karate (short of 'Karate') [49] and College Football (short of 'Football') [50] networks, and three artificial networks with different size generated by LFR model with specific parameters (i.e., $\mu = 0.25$, $\beta = 1.5$, $\gamma = 2.5$, $\langle k \rangle = 16$) [51]. Table 2 shows the experimental results, which suggest that these node similarity indexes based local topological information of the network can be competent for community detection. For real benchmark networks, the TJA-net behave better than the others, but for artificial networks, the proposed node similarity index is optimal to community detection. Note that the experimental results of the TJA-net is missed in the reference. Thus, we don't use it for the comparison analysis in artificial networks. Nevertheless, the comparison analysis of NMI based on these real benchmark networks and artificial networks suggest that the proposed method still maintains the high accuracy of community detection.

TABLE 2. Comparison analysis of NMI for diverse node similarity methods.

Algorithms	Real network		LFR network		
	Karate	Football	$n = 1000$	$n = 5000$	$n = 10000$
Jaccard	0.906	0.885	0.963	0.906	0.869
preference	0.947	0.902	0.833	0.805	0.772
Sørensen	0.914	0.938	0.979	0.937	0.901
TJA-net	1	0.927	–	–	–
LHN	0.945	0.922	0.997	0.995	0.980
Salton	1	0.922	0.986	0.963	0.951
HDI	0.890	0.938	0.870	0.861	0.830
ours	0.989	0.902	0.993	0.975	0.958

B. QUALITY ASSESSMENT OF THE COMMUNITY

Good community partitioning means that the nodes within the community are tightly connected, and that the external connections should be as sparse as possible. Thus, a complete quality assessment scheme should also consider the number of nodes in the community that are detected, the number of connections between nodes within the community, and the number of external connections. However, most of the existing quality assessment functions don't fully consider the above key points. For example, the famous modularity function [26] does not consider the influence of the number of nodes within the community on the modularity of whole community, and the triad participation ratio [2] does not consider the influence of the number of external connections. In order to make up for the deficiencies of the above method, we have defined a quality (evaluation) function Q as shown below,

$$Q = \frac{e_{int}}{n_{c_i} \cdot N_C} - \frac{e_{out}}{\sum_{l \in c_i} dev(l) - e_{int}}, \tag{11}$$

where e_{int} represents all the edges within a particular community and e_{out} represents the sum of all edges outside the community. The number of all nodes in the i^{th} community is represented by n_{c_i}. And the $dev(l)$ is the degree of node l, N_C is the number of all the communities in the division. Further, it can be understood as the averaging Q of all detected communities in the network partition,

$$Q = \frac{1}{|C|} \sum_{u \in C} Q_u. \tag{12}$$

where C is a collection of communities obtained through network division.

Fig.6 shows the values of Q obtained from the proposed method and the comparative ones (same in Fig.5 based on the email-EuAll and com-Amazon). For the email-EuAll, the general trend in Fig.6a is that the value of Q increases as φ increases, and for each specific value of φ, the proposed method also achieves a higher Q value. It also benefits from the fact that the edge weight influence factors are taken into account for evaluating node similarity. For $\varphi = 0.3$ and $\varphi = 0.7$, except of the fast greedy, other three methods behave approximate performance. Analogously, For the com-Amazon, the comparative analysis of Q in Fig.6b shows that with the increase of network size, the differences among methods aren't apparent. Nevertheless, the proposed method has achieved a fairly good competitiveness.

The above-mentioned method testing is performed in restricted to the influence factor of Jaccard similarity. In order to reflect the university of the proposed method, we conducted the additional method testing. Besides the Louvain, we additionally introduced three comparative methods, including the GN algorithm (short of 'GN') [50], other two algorithms (short of 'Li' and 'Chen') in the reference [52] that can be used for overlapping community detection. Three real benchmark networks (i.e., Karate, Football, and Netscience) are used to test these community detection methods. Fig.7 presents the experimental results that the comparison analysis of quality assessment Q for diverse community methods. For all real benchmark networks, the proposed method shows the highest values of Q, which reflects he university of the proposed method. Note that the differences among the quality assessment Q in restricted to diverse community detection methods become smaller when the network size increases (see in Fig.7c). The reason is that the size and structural complexity of the network are still within the acceptable range of these algorithms, and the overlapping phenomenon in the network is not serious.

C. TIME STRATEGY ASSESSMENT

For community detection in large-scale networks, how to control its reasonable time complexity is very challenging problem. Obviously, the time complexity is not only affected by the computational efficiency of community detection methods, it also relates to the data storage and retrieval in computational process. According to the above-mentioned framework of the proposed method, its overall time complexity is mainly reduced by adopting the parallel computing strategy based on the hash table based data storage and retrieval of node information to accelerate the computational efficiency of community detection methods. Naturally, the method testing also involves with the time strategy assessment, which is realized by quantifying the time complexity.

Fig.8 shows the comparison analysis of time performance of four methods. The experimental procedure is same to that in the comparative analysis in Fig.5 and Fig.6. More concretely, in Fig.8a, we can see that the running time of proposed method based on the email-EuAll is mostly lower than that of other three comparative ones in restricted to different influence factor φ of Jaccard similarity. But, only for $\varphi = 0.7$, the running time of the Louvain is little lower than that of the proposed method. It is due to the convergence of the above two algorithms at certain local time points, that is, they can quickly escape the influence of the local maximum and complete the community detection through the redistribution of nodes between communities. Furthermore, when φ increases from 0.3 to 0.8, the running time of proposed methods fluctuates in a relatively balanced way, which also benefits from the use of Fibonacci hash table [36], [53].

We turn to the running time testified based on the com-Amazon as shown in Fig.8b. It is found to be less compared

TABLE 3. Comparison analysis of time performance in more real benchmark networks.

Network	Ref.	Nodes	Edges	T1(s)	N1	T2(s)	N2
Karate	[49]	34	78	*na*	2	*na*	3
Dolphin	[54]	62	159	*na*	4	*na*	4
Football	[50]	115	613	*na*	11	*na*	6
Metabolic	[55]	453	1899	*na*	10	*na*	6
Jazz	[56]	198	2742	*na*	4	*na*	4
Email	[57]	1133	5451	6	10	11	14
M. Karplus	[56]	1166	13423	13	14	13	18
PPI-CP	[58]	4626	14801	11	27	19	35
Internet	[2]	11174	23409	27	44	37	41
Word Association	[5]	7207	32784	36	33	69	44
Collaboration	[59]	27519	116181	93	136	191	341
WWW	[19]	325729	1117563	261	1160	493	2017
Actor	[60]	82583	3666738	1422	327	7140	447
Actor weighted	[60]	82583	4475520	6242	277	3809	349

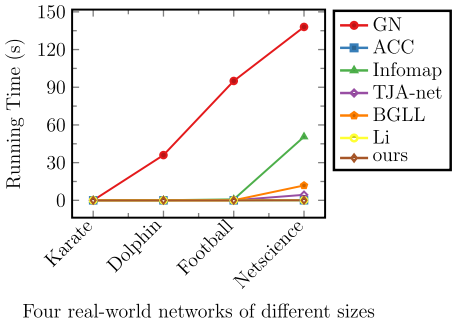


FIGURE 9. Comparison analysis of time performance of seven methods in four real benchmark networks.

with other three comparative methods. When φ is less than 0.7, the averaging running time of the proposed method is obviously the lowest. In particular, after a certain node hash table has been established, the community detection based node similarity only needs to use a constant-level time cost to query the node list when performing dynamic expansion, thereby reducing the running time (e.g., $\varphi = 0.5$ in Fig.8b). However, when the number of nodes increases continuously, especially when the edges among nodes increase by several orders of magnitude, it is inevitable that the running time will increase. For $\varphi = 0.8$, we can see that the running time of the fast greedy is significantly lower, which is also due to its extremely greedy strategy.

In order to study the time strategy assessment in more detail, we perform more comparison analysis of time performance by introducing more real benchmark networks, which is shown in Table 3. We simply illustrate the key symbols, that is, $T1$ and $T2$ are used respectively to represent the running time of the proposed method and the Newman, $N1$ and $N2$ indicate the number of the communities that they have detected, and *na* suggests a single run time with less than 10 milliseconds. Then, we specifically discussed the experimental results in Table 3. Except for the actor weight network (short of 'Actor weighted'), the running time of the proposed method is overall superior to that of the Newman. Especially, for the collaboration network (short of 'Collaboration'), the running time of the Newman increases by two times in comparison of that of the proposed method. Furthermore, the Newman detects more communities because it always tries to detect some extremely small community. For example, for the WWW network (short of 'WWW'), it detects the number of communities twice as much as the proposed method. And, according to the statistical analysis of the experimental results, the size of a part of communities is $5 - 10$ times smaller than that of the proposed method, which to some extent suggests that the quality assessment of the community is less accuracy. With these discussions, we can see that the hash table based data storage and retrieval strategy is reliable in reducing the time complexity of the parallel heuristic community detection method.

We also compare the time complexity of the proposed method with that of additional six benchmark methods, including the GN [50], the ACC [54], the Infomap [7], the TJA-net [46], the BGLL [44], the Li [52]. Fig.9 shows the comparison analysis of time performance for all methods based on four networks (i.e., Karate, Dolphin, Football and Netscience). The time performance is indicated by the averaging running time over 20 times experiments. As shown in Fig.9, in the network with small size (e.g., Karate), all methods have a close running time, however, in the network with large size (e.g., Netscience), the GN (138 seconds), the Infomap (50.6 seconds), the BGLL (11.8 seconds) and the TJA-net (4.4 seconds) have a higher running time. Moreover, the averaging running time of the proposed method, the ACC and the Li in restricted to all four networks is less than 1 second, suggesting they have optimal time performance in community detection.

We summarized the time complexity of the proposed method based on the whole process framework. The parallel computing strategy of the hash table based on data storage and retrieval can complete query within a constant time level regardless of the network size. As a consequence, the running time is mainly induced by calculating node similarities and merging nodes into communities. Remove the running time used to query a certain node, the computational time consumption of one node and its k neighbor nodes is $O(k)$, where k is the average degree of nodes in the network. Thus, for a network with n nodes, the time complexity of calculating node similarities is $O(nk)$. Then, the time complexity required to merge the current community with a node that has a determined similarity is $O(1)$ CPU time, and the time complexity in the community consolation is $O(n)$. In summary, the overall time complexity of the proposed method is $O(nk) + O(n)\ O(nk)$. Considering that these real networks are usually very sparse (i.e., the average degree k is very smaller comparing with the network size n), the time complexity of the proposed method is approximately in a linear relationship with the network size.

D. HASH BEHAVIOR ASSESSMENT

The overall performance of parallel computing strategy is affected by efficiently dealing with the hash table based on

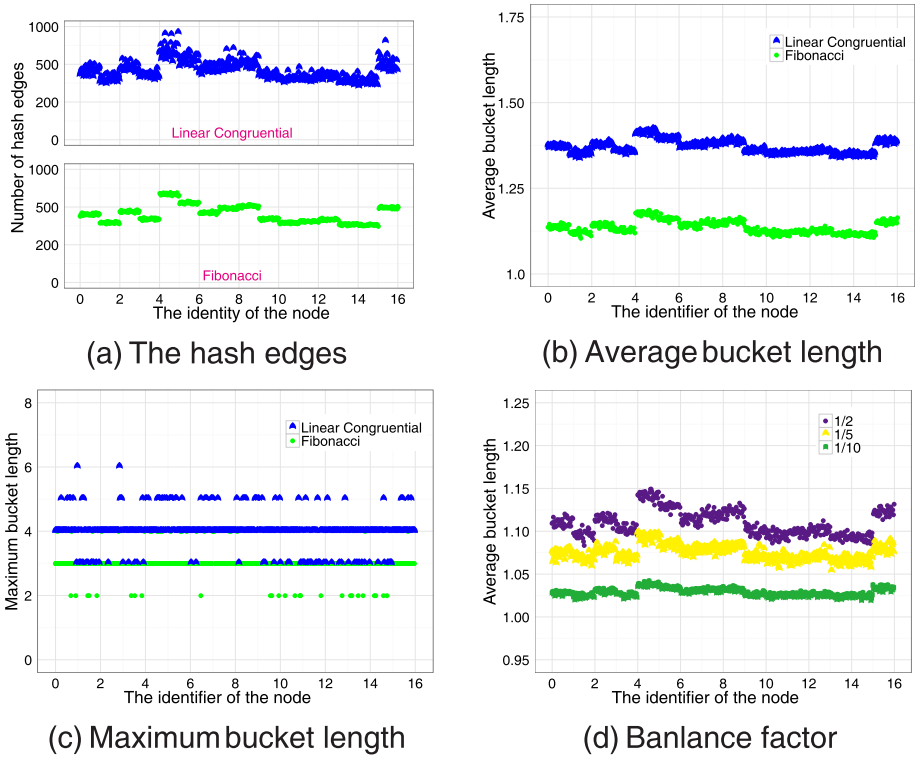


FIGURE 10. Comparison analysis of hash behavior using the Fibonacci hash function and the linear congruential hash function based on com-Amazon dataset.

data storage and retrieval. In order to guarantee the high overall performance of parallel computing strategy, the hash function plays an important role. For that, the method testing also includes the hash behavior assessment. Several types of hash functions have been tested, including the Fibonacci hash function, the concatenated hash function and linear congruential hash function [53]. Combined with the parallel computing strategy, we firstly determine that the Fibonacci hash function and the linear congruential hash function can achieve a better overall performance. Thus, we further make a control test for these two hash functions based on the com-Amazon, of which the experimental results are shown in Fig.10.

Firstly, as shown in Fig.10a-Fig.10c, the load balance effect of hash table is compared. Each node is assigned a uniform graph vertex partition. The incident edge is assigned to the corresponding node storage hash table. The bucket in the hash table of each node will be partitioned again according to the thread of the node. By processing like this way, the data conflict during the query can be controlled to a very small extent. Fig.10a illustrates the number of hash edges delegated to each thread. Although the load of each node is uniform and is determined by Eq.(6), the Fibonacci hash function provides better load balancing within each node. The relatively small average bucket length in Fig.10b confirms the result. And, only the case of more than or equal to one edge is considered when calculating the average bucket length. In Fig.10c, it is clearly observed that the maximum bucket length is 2.9 and 6.2, respectively.

Then, the balance factor [37] is another very important performance evaluation parameter. More precisely, it is a rebalancing of performance and computing resources. A reasonable balance factor can reduce the hash table conflicts, but it may take up more memory space. Fig.10d compares the average bucket length of each thread using different balance factors. As expected, a larger balance factor will result in a larger bucket length. When the balance factor is $1/10$, the average bucket length is close to 1, which also greatly improves the hit rate in the query. It can be seen from this experiment that the smaller balance factor may bring better performance when the memory allows. And it also provides some useful reference information for us in configuring the running mechanism of parallel environment. In this paper, choose $1/5$ as the experimental standard, which is also a balance between time efficiency and memory consumption.

V. DISCUSSION

In this paper, a new evaluation criteria of node similarity is adopted to detect the community structure of network. The core idea is to divide the nodes into different types with their neighbors as the bridge, and assign different weight influence factor to the edges. The effectiveness of the node similarity criteria and parallel heuristic community detection method is proved by both the NMI score and the quality assessment. Also, the time complexity of parallel heuristic community detection method is analyzed and tested by the comparison analysis of other community detection methods

based a number of real benchmark networks. In the following, we simple discuss these aspects.

In the method testing of NMI score, we first compared the proposed method with some classical community detection methods. The experimental results show that the NMI calculated by the proposed method is generally better than others. We analyze the potential reason that when calculating the node similarity, the common neighbors of the nodes and their corresponding neighbors are fully considered, so the acquisition of local topological information of each node pair is superior to other methods that do not fully consider such characteristics. According to Fig.5, the overall performance of the proposed method is better than Newman (e.g., the results obtained by the algorithm in this paper is increased by 8.39% in Fig.5a and is increased by 1.95% in Fig.5b when $\varphi = 0.5$), because there is no so-called resolution problem in this paper, that is, the community structure with small number of nodes can be detected. The fast greedy algorithm is developed based on the Newman (e.g., 14.43% in Fig.5a when $\varphi = 0.3$ and 15.89% in Fig.5b when $\varphi = 0.1$), so it has similar properties. Compared with the Louvain (e.g., 7.82% in Fig.5a and 8.27% in Fig.5b when $\varphi = 0.5$), because the proposed method is based on node attributes, so it will not fall into the trap of local maximum value and can obtain greater benefits of modularity. The comparison analysis of all methods in Table 2 further proves the effectiveness of the proposed method. For the artificial networks generated via LFR model, the performance of all methods is affected by the increasing complexity of its network structure. However, the proposed method is consistent with the LHN, because it takes into account the topology and the properties of the nodes, and even significantly better than most other algorithms (e.g., compared with Salton, which is second only to our algorithm, the results obtained by our algorithm are improved by 1.25% when $n = 5000$.

In the method testing of quality assessment, we introduced a new quality evaluation function to make it more accurate to judge the resulting community structure. In Fig.6, compared with other methods, the proposed method has obtained a higher Q value on the whole. In particular, it has obtained the largest Q value that is more obvious than others when $\varphi = 0.5$ in Fig.6a (e.g., the results obtained by our algorithm are improved by 3.53% compared to the Louvain algorithm). This case also benefits from the fact that the proposed method takes into account different weight influence factors of the edges among nodes. In Fig.7b, because the proposed method considers the different neighbor types among nodes and assigns different weight influence factors for the affinity among nodes, its performance behaves better than the Louvain and the Li in the assessment of community detection. In fact, these two methods may fall into local maximums, and their modularity will not increase without external forces. However, the proposed method considers the network topology independent from the selection of the initial node, and only involves with the node similarity matrix in the process of merging community, so that it can achieve the better community detection quality (e.g., the results obtained by our algorithm are improved by 3.49% compared to the Li algorithm in Fig.7b).

In the method testing of time complexity, we used a dynamic hash table based data storage and retrieval to manage the information of node triples. With the increase of φ, it can be seen in Fig.8 that the running time of the proposed method using the parallel computing strategy is relatively smaller. In fact, after the establishment of the hash table of node information, the data retrieval of node similarity calculation can be completed in the constant-level time when performing the dynamic expansion of the community. As a result, the running time is greatly reduced (for $\varphi = 0.5$ in Fig.8b, compared with the Newman algorithm, the results obtained in this paper reduce the time consumption by 1.94%). Considering more real benchmark networks, we compared the proposed method with the Newman and the experimental results are shown in Table 3. Except for the Actor weighted, the proposed method is superior to the Newman in time efficiency, which may be due to the serious overlapping phenomenon of the network (e.g., in Actor network, the results obtained by the algorithm in this paper reduce the time consumption by 80.08%). Fig.9 is a further comparison analysis of time performance. Compared with the ACC based on maximum sub-graph division and the clustering factor of nodes (these two stages consume little time compared to other algorithms), the proposed method is consistent with its time performance. Compared to the TJA-net that has an extremely high community detection accuracy based on the idea of label propagation, the running time is sightly higher than that of the proposed method due to the time consumption of a node fine-tuning and merging community in the final stage (e.g., in Netscience network, the results obtained by our algorithm reduce the time consumption by 98.86%). In addition, the Fibonacci hash function is used to calculate the hash value of nodes in this paper, which further reduces the probability of conflict in data retrieval.

VI. CONCLUSION

In this paper, we have proposed a heuristic parallel community detection method node similarity, based on the core idea that the higher the similarity between nodes, the greater the tendency of the community to form. An evaluating method of node similarity is also introduced by assigning different edge weight influence factors based on the impact of different neighbor types of nodes on node similarity. That is, by using the common neighbors of the node pair as bridge, the neighbor nodes of such common neighbors that affect the similarity calculation are also taken into account to increase more local topological information and the interactions of the node pair. In the process of evaluating node similarity, because we need to query the edge types and assign its corresponding weight influence factor, we developed a parallel computing strategy by the hash table based data storage and retrieval. The strategy hashes the edge information into a ternary structure that can be merged according to the same starting node. In addition,

it is stored in very low conflicts in a hash bucket that can be retrieved in the context of a constant-level time complexity.

Base on the method testing and evaluation of four aspects, the experimental results show that the proposed method is very suitable to detect the community structure in complex networks with a large-scale size. More concretely, the community detection accuracy is evaluated by the NMI and quality function. Base on such metrics, we performed the comparison analysis of the proposed method with other community detection methods in restricted to diverse real benchmark networks and artificial networks. On one hand, we considered the detection of the overlapping nodes and the importance of the central nodes, and thus firstly divide the original network into communities with Jaccard similarity. Then, we further performed the method testing based on these community-scale networks. Compared with the fast greedy, the proposed algorithm in this paper improves the NMI score by 14.43% in Fig.5a when $\varphi = 0.3$ and 15.89% in Fig.5b when $\varphi = 0.1$. Compared with the Louvain, the proposed algorithm improves the quality score by 3.53% in Fig.6a when $\varphi = 0.5$. On the other hand, to verify the university of the proposed method, we also used the original networks for method testing. Compared with Salton, the NMI score obtained by our algorithm based on LFR model is improved by 1.25% when $n = 5000$ in Table 2 and the quality score is improved by 3.49% compared to the Li algorithm in Fig.7b. Therefore, it can be seen from the experimental results that the method testing based on these two types of networks both show the community detection accuracy of the proposed method is higher than most of the comparative methods. Furthermore, the time performance is evaluated by the running time. The experimental results in diverse real benchmark networks also show the lower running time in comparison of other community detection methods (e.g., in Actor network, the results obtained by our algorithm reduce the time consumption by 80.08% compared with Newman in Table 3 and reduce the time consumption by 98.86% for Netscience network in Fig.9 compared with TJA-net that has an extremely high community detection accuracy). And, we have presented a simple analysis of time complexity of the proposed method, which is approximately in a linear relationship with the network size.

Community detection in complex network is an open issue, and the choice of perspective determines the differences in research methods. The node similarity based detection method is designed based on the nodes of various roles in the network, which determines the intimate relationships among them. It is reasonable to divide or classify the nodes according to these intimate relationships. For example, the impact of a central node on its neighbors should be greater than that of marginal nodes. In addition, while ensuring the quality of community detection, how to reduce the consumption of computing time is also a very challenging problem. This paper only makes some attempts on the data storage strategy, to some extent, this still depends on the topology of the network. It should be noted that an excellent method in robustness should take into account the common effects of multiple factors. The proposed method can be extended to larger scale dynamic network detection with overlapping phenomenon. It is believed that combining it with the current popular distributed processing system will result in a tremendous increase in both computational efficiency and accuracy, and this needs to be further explored and research.

REFERENCES

[1] 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, 2009.

[2] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[3] S. Fortunato and C. Castellano, "Community structure in graphs," in *Computational Complexity: Theory, Techniques, and Applications*. Cham, Switzerland: Springer, 2012, pp. 490–512.

[4] Y. Pan, D. Li, J.-G. Liu, and J.-Z. Liang, "Detecting community structure in complex networks via node similarity," *Phys. A, Stat. Mech. Appl.*, vol. 389, no. 14, pp. 2849–2857, 2010.

[5] J. Yang, J. McAuley, and J. Leskovec, "Coprime coarray interpolation for DOA estimation via nuclear norm minimization," in *Proc. IEEE 13th Int. Conf. Data Mining (ICDM)*, May 2013, pp. 1151–1156.

[6] J. Shao, Z. Han, Q. Yang, and T. Zhou, "Community detection based on distance dynamics," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1075–1084.

[7] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.

[8] Z. Zhuo, S.-M. Cai, M. Tang, and Y.-C. Lai, "Accurate detection of hierarchical communities in complex networks based on nonlinear dynamical evolution," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 28, no. 4, p. 043119, 2018.

[9] J. Shao, Z. Zhang, Z. Yu, J. Wang, Y. Zhao, and Q. Yang, "Community detection and link prediction via cluster-driven low-rank matrix completion," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3382–3388.

[10] C. Zhe, A. Sun, and X. Xiao, "Community detection on large complex attribute network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2041–2049.

[11] Y. Lei and S. Y. Philip, "Cloud service community detection for real-world service networks based on parallel graph computing," *IEEE Access*, vol. 7, pp. 131355–131362, 2019.

[12] R. K. Behera, D. Naik, B. Sahoo, and S. K. Rath, "Centrality approach for community detection in large scale network," in *Proc. 9th Annu. ACM India Conf.*, 2016, pp. 115–124.

[13] G. Ren and X. Wang, "Epidemic spreading in time-varying community networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 24, no. 2, p. 023116, 2014.

[14] M. E. Newman, "Equivalence between modularity optimization and maximum likelihood methods for community detection," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 94, no. 5, p. 052315, 2016.

[15] Y. Cui and X. Wang, "Uncovering overlapping community structures by the key bi-community and intimate degree in bipartite networks," *Phys. A, Stat. Mech. Appl.*, vol. 407, pp. 7–14, Aug. 2014.

[16] Y. Cui and X. Wang, "Detecting one-mode communities in bipartite networks by bipartite clustering triangular," *Phys. A, Stat. Mech. Appl.*, vol. 457, pp. 307–315, Sep. 2016.

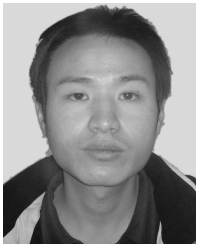
[17] X. Wang and X. Qin, "Asymmetric intimacy and algorithm for detecting communities in bipartite networks," *Phys. A, Stat. Mech. Appl.*, vol. 462, pp. 569–578, Nov. 2016.

[18] G. Kechagias, G. Tzortzis, G. Paliouras, and D. Vogiatzis, "A parallel algorithm for tracking dynamic communities based on apache flink," in *Proc. 10th Hellenic Conf. Artif. Intell.*, 2018, p. 9.

[19] H. Lu, M. Halappanavar, and A. Kalyanaraman, "Parallel heuristics for scalable community detection," *Parallel Comput.*, vol. 47, pp. 19–37, Aug. 2015.

[20] R. Aktunc, I. H. Toroslu, M. Ozer, and H. Davulcu, "A dynamic modularity based community detection algorithm for large-scale networks: DSLM," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2015, pp. 1177–1183.

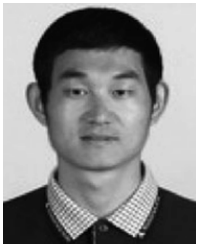
[21] D. Palsetia, W. Hendrix, S. Lee, A. Agrawal, W.-K. Liao, and A. Choudhary, "Parallel community detection algorithm using a data partitioning strategy with pairwise subdomain duplication," in *Proc. Int. Conf. High Perform. Comput.* Cham, Switzerland: Springer, 2016, pp. 98–115.
[22] D. E. Knuth, *The Art of Computer Programming*, vol. 3. London, U.K.: Pearson Education, 1997.
[23] M. Bartík, T. Beneš, and P. Kubalík, "Design of a high-throughput match search unit for lossless compression algorithms," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0732–0738.
[24] P. K. Gopalan, C. Wang, and D. Blei, "Modeling overlapping communities with node popularities," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2850–2858.
[25] J. Li, X. Wang, and J. Eustace, "Detecting overlapping communities by seed community in weighted complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 23, pp. 6125–6134, Dec. 2013.
[26] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.
[27] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, 2007.
[28] J. Eustace, X. Wang, and Y. Cui, "Community detection using local neighborhood in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 436, pp. 665–677, Oct. 2015.
[29] X. Wang and J. Li, "Detecting communities by the core-vertex and intimate degree in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 10, pp. 2555–2563, May 2013.
[30] J. Eustace, X. Wang, and Y. Cui, "Overlapping community detection using neighborhood ratio matrix," *Phys. A, Stat. Mech. Appl.*, vol. 421, pp. 510–521, Mar. 2015.
[31] Y. Cui, X. Wang, and J. Eustace, "Detecting community structure via the maximal sub-graphs and belonging degrees in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 416, pp. 198–207, Dec. 2014.
[32] R. S. Burt, "Positions in networks," *Social Forces*, vol. 55, no. 1, pp. 93–122, 1976.
[33] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Stat. Appl. Genet. Mol. Biol.*, vol. 4, no. 1, 2005.
[34] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens, "Graph nodes clustering with the sigmoid commute-time kernel: A comparative study," *Data Knowl. Eng.*, vol. 68, no. 3, pp. 338–361, 2009.
[35] H. Tian, Y. Chen, C.-C. Chang, H. Jiang, Y. Huang, Y. Chen, and J. Liu, "Dynamic-hash-table based public auditing for secure cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 701–714, Sep./Oct. 2015.
[36] H. Lang, V. Leis, M.-C. Albutiu, T. Neumann, and A. Kemper, "Massively parallel NUMA-aware hash joins," in *In Memory Data Management and Analysis*. Cham, Switzerland: Springer, 2015, pp. 3–14.
[37] R. A. Dunlap, *The Golden Ratio and Fibonacci Numbers*. Singapore: World Scientific, 1997.
[38] 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, p. 046110, 2008.
[39] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 555–564.
[40] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, 2015.
[41] 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, p. 066133, 2004.
[42] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, p. 066111, 2004.
[43] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.
[44] R. Shang, W. Zhang, L. Jiao, R. Stolkin, and Y. Xue, "A community integration strategy based on an improved modularity density increment for large-scale networks," *Phys. A, Stat. Mech. Appl.*, vol. 469, pp. 471–485, Mar. 2017.
[45] T. A. Sorensen, *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons* (Biologiske Skrifter), vol. 5, E. Munksgaard, Ed. Tokyo, Japan: CiNii Articles, 1948, pp. 1–34.
[46] R. Shang, H. Liu, L. Jiao, and A. M. G. Esfahani, "Community mining using three closely joint techniques based on community mutual membership and refinement strategy," *Appl. Soft Comput.*, vol. 61, pp. 1060–1073, Dec. 2017.
[47] 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, p. 026120, 2006.
[48] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.
[49] M. E. J. Newman, "Spectral methods for community detection and graph partitioning," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 88, no. 4, p. 042822, 2013.
[50] 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, Apr. 2002.
[51] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 9, p. P09008, 2005.
[52] J. Li, X. Wang, and Y. Cui, "Uncovering the overlapping community structure of complex networks by maximal cliques," *Phys. A, Stat. Mech. Appl.*, vol. 415, pp. 398–406, Dec. 2014.
[53] X. Que, F. Checconi, F. Petrini, and J. A. Gunnels, "Scalable community detection with the Louvain algorithm," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2015, pp. 28–37.
[54] Y. Cui, X. Wang, and J. Li, "Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient," *Phys. A, Stat. Mech. Appl.*, vol. 405, pp. 85–91, May 2014.
[55] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, p. 651, 2000.
[56] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multi-step greedy algorithm and vertex mover refinement," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 77, no. 4, p. 046112, 2008.
[57] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 2, p. 025101, 2004.
[58] V. Colizza, A. Flammini, A. Maritan, and A. Vespignani, "Characterization and modeling of protein–protein interaction networks," *Phys. A, Stat. Mech. Appl.*, vol. 352, no. 1, pp. 1–27, 2005.
[59] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, 2001.
[60] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.



QIANG ZHOU received the B.S. and M.S. degrees in computer science and technology from Guizhou University, Guiyang, China, in 2012. He is currently pursuing the Ph.D. degree in computer software and theory with the University of Electronic Science and Technology, Chengdu, China.

Since 2014, he has been studying in the Institute of Fundamental and Frontier Science, University of Electronic Science and Technology. His main interests include social networking, big data analytics, complex network community detection, and recommendation algorithm research.

His main awards and honors include national scholarships, national motivational scholarships, and provincial outstanding graduates.



SHI-MIN CAI received the B.S. degree in electrical engineering from the Hefei University of Technology, in 2004, and the Ph.D. degree in circuit and systems from the University of Science and Technology of China, in 2009.

He currently serves as an Associate Professor of the University of Electronic Science and Technology of China. At present, he has published nearly 100 high-level academic articles, including nearly 70 SCI articles, nearly 400 SCI quotations, and completed more than 10 national projects supported by the National Natural Science Foundation of China and the Military Commission for Science and Technology. He is interested in complex network theory and its application for mining and modeling of real large-scale networked systems, time series analysis, and personalized recommendation systems.

YI-CHENG ZHANG received the B.S. degree from the Physics Department, University of Science and Technology of China, Hefei, China, in 1980, the M.S. degree in physics from the Graz University of Technology, Graz, Austria, in 1981, and the Ph.D. degree in physics from University of Rome La Sapienza, Roma, Italy, in 1984.

He is currently a Full Professor with the University of Fribourg, Switzerland, and also a Distinguished Professor with the University of Electronic Science and Technology of China. So far, he has published more than 130 SCI articles, in which 35 published in *Physics Reports*, PNAS, and *Physical Review Letters*. He got more than 24297 times cites according to Google Scholar. His main research interests include complex networks, cloud computing, and big data processing.