Published in "IEEE Access 7(): 184145–184159, 2019"
which should be cited to refer to this work.

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 com-munity 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

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,

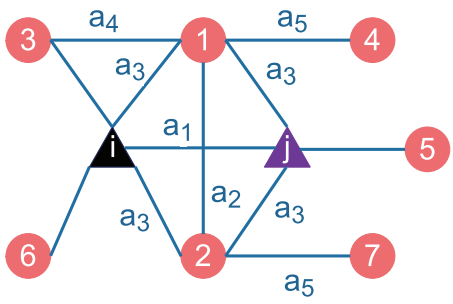
$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}, \tag{1}$$

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.

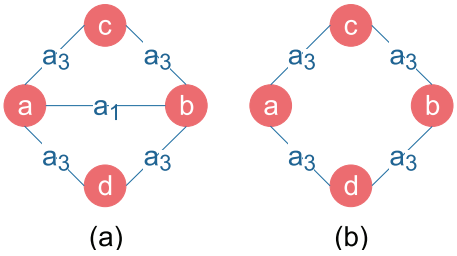


FIGURE 2. An example of node similarity comparison using the proposed method and Jaccard similarity. For a pair of nodes *a* and *b*, according to Eq.(4) and Eq.(5), (a) the nodes similarity is respectively 0.68 and 0.5; (b) the nodes similarity is respectively 0.6 and 1.

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),

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

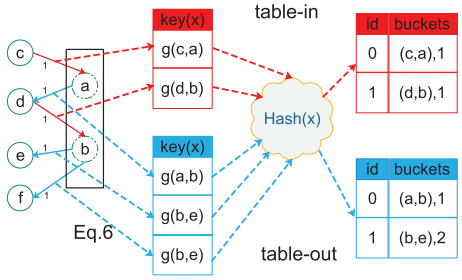


FIGURE 3. A hash table based data storage and retrieval strategy for storing node information.

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.

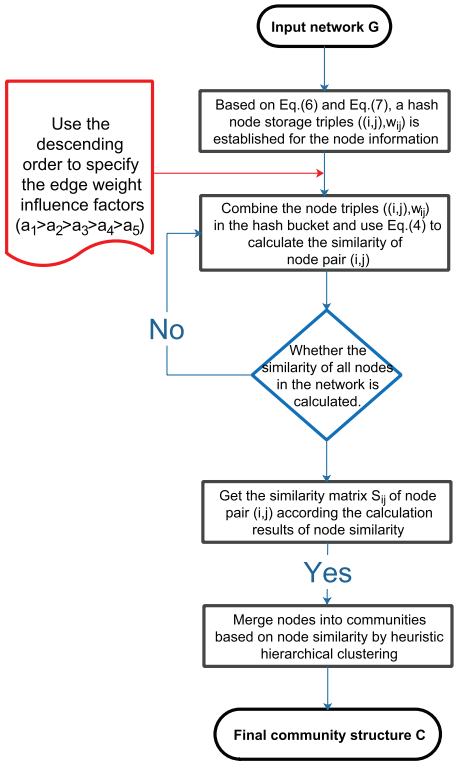


FIGURE 4. This flow chart of entire algorithm principal frame of parallel heuristic community detection method.

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