IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON GREEN CLOUD AND FOG COMPUTING: ENERGY EFFICIENT AND SUSTAINABLE INFRASTRUCTURES, PROTOCOLS AND APPLICATIONS

Received July 26, 2017, accepted August 13, 2017, date of publication August 22, 2017, date of current version September 19, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2740962

Efficient Vector Influence Clustering Coefficient Based Directed Community Detection Method

XIAOLONG DENG[1], JIAYU ZHAI[2], TIEJUN LV[3], (Senior Member, IEEE), AND LUANYU YIN[4]

[1]Key Laboratory of Trustworthy Distributed Computing and Service of Education Ministry, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]International School, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[4]China Academy of Social Management, Beijing Normal University, Beijing 100875, China

Corresponding author: Xiaolong Deng (shannondeng@bupt.edu.cn)

This work was supported in part by the Philosophy and Social Science Project of Education Ministry under Grant 15JZD027, in part by the National Culture Support Foundation Project of China under 2013BAH43F01, in part by the National 973 Program Foundation Project of China in social network analysis under Grant 2013CB329605, and in part by the National Natural Science Foundation of China under Grant 91224008-14 through the Joint-Operated Project.

ABSTRACT Community detection algorithms are important for determining the character statistics of complex networks. Compared with the conventional community detection algorithms, which always focus on undirected networks, our algorithm is concentrated on directed networks such as the WeChat moments relationship network and the Sina Micro-Blog follower relationship network. To address disadvantages such as lower execution efficiency and higher deviation of precision that current directed community detection algorithms always have, we propose a new approach that is based on the triangle structure of community basis and modeled on the local information transfer process to precisely detect communities in directed networks. Based on the directed vector theory in probability graphs and the dynamic information transfer gain (ITG) of vertices in directed networks, we propose the novel ITG method and the corresponding target optimal function for evaluating the partition quality in a community detection algorithm. Then, we combine ITG and the target function to create the new community detection algorithm ITG-directed weighted community clustering for directed networks. With extensive experiments using artificial network data sets and large, real-world network data sets derived from online social media, our algorithm proved to be more accurate and faster in directed networks than several traditional, well-known community detection methods, such as FastGN, order statistics local optimization method, and Infomap.

INDEX TERMS Community detection algorithm, information transfer gain, target function optimization, community detection in directed network.

I. INTRODUCTION

Research into large-scale social networks is becoming increasingly important. Due to the convenience of connection to each other, the scale of online social networks is enlarging at an unprecedented rate. By the end of the last day of 2015, for example, the number of global Twitter users was more than 500 million, including 200 million active users. WeChat users number was more than 600 million, with 400 million active users.

Community detection in these large-scale social networks plays an important role in the study of topologies and the architecture of networks. Because of the huge quantity of vertices and edges in a large-scale network and its complex structure, common traditional graph analysis approaches cannot perform the research such as layered architecture analysis and knowledge attaining within reasonable execution times. Therefore, the high processing efficiency and accurate results of community detection algorithms are necessary to detect potential community structures from the huge directed complex network.

To solve the problem of traditional community detection algorithms in large-scale social networks and directional social networks such as WeChat without accurate simulation models and high algorithm executing efficiency and precision, this article starts with the triangle group, which is the basic structure of the community and modelling on the local impact of vertex in networks. By using directional vector information spread, probability calculation theory

and probability graph theory modelling those vertices with huge influence on directed social networks, this paper constructs directed clustering coefficients for vertexes. We also construct the target function for measuring the efficiency of community detection and distributed parallel community detection algorithm of our model for large-scale social networks. Ultimately, by conducting experiments on large-scale artificial network datasets and real network datasets, the precision and novelty of our algorithm is verified.

In this paper, section 2 shows the current related work on community detection algorithms in large-scale networks. Section 3 introduces our ITG model in detail and the ITG-based directed network community detection algorithm ITG-DWCC. Section 4 shows the comparison of experimental results of the ITG-DWCC algorithm and other classical algorithms in artificial network datasets and real-world network datasets. Section 5 gives the conclusion.

II. RELATED WORK

In networks, communities are divisions composed of vertices and edges, which are called Groups and Clusters. Community structure has the basic characteristic that vertices in the same community are connected closely with each other and vertices in different communities are connected sparsely. Furthermore, information spreads faster inside a community than among different communities.

Based on different analysis targets, current community detection algorithms can be divided into four groups: the hierarchical clustering approach, the matrix spectrum analysis approach, the edge based approach, and the maximum clique based approach [1]. Focusing on large-scale networks, there are three categories of community detection algorithms: the modularity value optimization based method, the random walk based method and the overlapping community detection based method.

A. MODULARITY VALUE OPTIMIZATION-BASED METHOD

This method attempts to bring the idea of small-scale community detection algorithms, which are based on modularity optimization, into large-scale community detection. Through the optimization of modularity, a fine-grained community detection result is obtained. A function of modularity Q is proposed by Newman and Girvan [2], which is defined as:

$$Q = \sum_{r} (e_{rr} - \alpha_r^2) \tag{1}$$

In formula (1), e_{rr} represents the total internal edge number of community r, and α_r represents the sum of internal edge numbers and external edge numbers of the community. If we regard a community as a sub-graph, the edge number of a corresponding random graph model would be less than the actual edge number. The better structure a community has, the larger is the value of the modularity function.

These kinds of approaches are mostly optimization methods of modularity in large-scale network community detection. The classical representative algorithm is FastGN. In 2004, Newman proposed an algorithm named FastGN (FN) [3] that is based on modularity optimization in the edge exchanging process. This algorithm makes use of the Q value gain in each edge exchanging process among different communities to find the optimal direction of modularity Q. However, FastGN has low efficiency when the network scale is very large. Later, Clauset and Newman used heap structure for improvement and proposed the CNM algorithm [4]; its complexity is nearly linear to network scale in large-scale networks.

In 2008, Blondel proposed a fast algorithm named Louvain [5]. Louvain also used modularity optimization to process network community detection. When modularity converged to a maximum value, it would stop the detection process. By the end of 2014, it was regarded as the fastest algorithm in community detection in large-scale networks. An article in WWW 2014 [6] by Arnau Prat-Pérez considered this algorithm quickly decreasing in performance when network scale increased, which shows the algorithm needed more study in 2014.

Another famous algorithm is the LPA (Label Propagation Algorithm) [7] series-based large-scale community detection algorithm. The time complexity of LPA is $o(n^2)$ where n represents the node number of the network. Compared to other complex machine learning algorithms, LPA has lower complexity and better clustering efficiency. In 2007, Raghvan improved LPA by providing the RAK algorithm [8], which was based on community detection operation with an approximate linear direct ratio when network scale increased. Through the predefined target function, it simplifies the complexity of LPA and uses network structure as a guide to detect community structure. Its result in the Karate Club network and the American University football network shows the good performance of RAK. However, there are some special drawbacks of the RAK algorithm in experiments in benchmark networks, and it needs improvement. In 2010, Gregory improved RAK by providing COPRA, (Community Overlap PRopagation Algorithm) [9] which is an algorithm that focuses on mining overlapping communities. In the COPRA algorithm, every single node has a number of community labels. Furthermore, the propagation process of COPRA includes multi-information of community, which contributes to the increase in execution time cost in each iteration process. While there are lots of overlapping communities, it results in the random selection of fault labels.

In 2011, Jin *et al.* [10] proposed the approximate linear rapid LPA hereditary algorithm FNCA, which was based on local detection optimization. This algorithm improved the ending condition of the fifth iteration in LPA, which saves 20% of iterations. In 2011, Cordasco and Gargano [11] proposed the semi-synchronous LPA algorithm. It improved execution efficiency by network vertex parallel colouring technology with synchronous and asynchronous modelling. It was useful for large-scale networks but limited in the solution of modularity calculations. When the graph scale

is quite large, it cannot find small-scale and well-defined communities.

B. RANDOM WALK BASED METHOD

Random walk based methods have the characteristic that information is spread easily in the internal, high-density community. Different from modularity optimization based methods, this type of approach focuses on the process of information propagation or some physical element permeation while it attains the community structure fast and effectively. Some classical representative algorithms are as follows.

In 2006, Pons proposed the random walk community detection algorithm Walktrap [12], which was based on the similarity of nodes in large-scale networks. By using the definition of Euclidean distance for the distance among different communities, it has good time complexity; the best time complexity is $o(mn^2)$ and the ordinary time complexity is $o(n^2 \log n)$. In 2008, Rosvall summarized the introduction of random walk based community detection algorithms in detail and set up a model for the probability of information flowing in different nodes by using the information entropy function of Information Theory and presented the Infomap algorithm [13]. Through the comparison of the experimental results from many large-scale scientist cooperation networks and the LFR [14] standard dataset, the Infomap algorithm has been proven to better perform than some overlapping community detection algorithms.

C. OVERLAPPING COMMUNITY DETECTION METHOD

The overlapping community detection method has a different construction than the previous algorithms. The representative algorithms are as follows. In 2005, Palla proposed the Clique Percolation Method algorithm (CPM) [15], which was based on the characteristic that the edge of an internal community has tight connections and is easy to form a clique, which finally consists of a community.

However, the CPM algorithm is very strict with the limitation of connections among cliques, which causes high time complexity. In 2010, Ahn proposed the Link Clustering Algorithm (LCA) [16] by setting up a model based on edges rather than on nodes. In the calculating process, it used the Jaccard coefficient to calculate the similarity of connected edges, which makes the existence of overlapping communities natural. In 2011, Filippo presented a measuring function, which used a Q value based significance function as the detection object function and invented the Order Statistics Local Optimization Method (OSLOM) [17]. The OSLOM algorithm is the first algorithm for community detection in directed weighting edge networks. LCA is also constructed based on an optimization significance function value. In 2013, Yang and Leskovec improved the LPA algorithm by optimizing group nodes for community attached relation target functions and presented the BigClam algorithm [1]. By redefining the overlapping community in different communities, this algorithm produces good results in large-scale networks.

However, all the algorithms above do not focus on directed large-scale networks for community detection and have long execution times and low accuracy, so it is necessary to construct a highly efficient algorithm to make improvements.

III. VECTOR INFLUENCE CLUSTERING-BASED DIRECTED NETWORK COMMUNITY DETECTION ALGORITHM

In an actual social network relationship, two friends of someone can be friends of each other, and this attribute can be called the clustering characteristic of networks [18]. The network average clustering coefficient reflects the microscopic clustering characteristics of a network and has become an important index of adjacent nodes that connect closely. The node clustering coefficient definition of a network is: In a network with N nodes, one node i has k_i edges connected to it and other nodes, i.e., node i has k_i, neighbours. If among the k_i nodes, there are E_i edges, the clustering coefficient of node i is:

$$C_{\mathrm{i}} = \frac{2E_{\mathrm{i}}}{k_{\mathrm{i}}(k_{\mathrm{i}} - 1)} \tag{2}$$

From a geometric perspective, C_{i} in formula (2) can be defined equally as:

$$C_{\mathrm{i}} = \frac{|TriangleStructure(i)|}{|TripleStructure(i)|} \tag{3}$$

In formula (3), the triple is the structure that connected with node i, includes node i itself and two other nodes, and there exists at least two edges among node i and the other two nodes:

The network average clustering coefficient is defined in formula (4):

$$C_{G=(V,\varepsilon)} \equiv \frac{1}{N} \sum_{v_i \in V, i=1}^{N} \frac{2E_{\mathrm{i}}}{k_{\mathrm{i}}(k_{\mathrm{i}} - 1)} \tag{4}$$

The network average clustering coefficient is used for measuring connection density of triangle structures in the network. While the network has a larger proportion of triangle structures, these triangle structures belong to tighter community structures in internal connections and the network average clustering coefficient is larger. The clustering coefficient of the whole network has the value scope of $0 \leq C_{G=(V,\varepsilon)} \leq 1$.

In fact, in many types of social networks, the probability that a friend of user u is a friend of friends of user u verges to a constant as the network scale N increases [18]. When $N \to \infty$, $C_{G=(V,\varepsilon)} = O(1)$ is converged to a nonzero constant, which reflects the characteristic that "things of one kind come together."

A. VECTOR INFLUENCE-BASED CLUSTERING COEFFICIENT MODEL

When setting up a model for a traditional social network detection algorithm, network is often processed as an unweighted and undirected graph, ignoring the direction of edges. However, in online social networks, edge direction

always contains important information. Online social networks have the characteristic that the important nodes (such as opinion leaders) are always information propagation originators. In this paper, based on the classic Probabilistic Graphical Model (PGM) theory from the Turing Award winner Pearl, we extract the direction of information propagation edge between different social network nodes to a directed vector and propose a vector influence clustering coefficient model with both information propagation direction and information propagation probability.

The PGM theory uses graph structure to represent the joint probability distribution of variables. In recent years, it has become the research hub for uncertainty reasoning solutions. The PGM representative theory combines knowledge of probability theory and graph theory. In graph theory, the relationship of random variable dependency can be represented and provides an effective representing frame for statistical multivariable modelling.

The PGM theory can always be classified by two conditions: (1) edge based directed or undirected attributes, and (2) abstraction based different levels. Based on directed or undirected attributes of edges to classify PGM, there are three classes: (1) a directed graph model called the Bayesian network (BN) [19], where the network structure is a directed acyclic graph; (2) an undirected graph model called the Markov Network (MN) [19], where the network structure is an undirected graph; and (3)a local directed model including both directed edges and undirected edges, that is composed of a Conditional Random Field (CRF) and a Chain Graph (CG). Based on the different levels of abstraction, there are two classified classes: (1) a random variable based probability graph model such as BN, MN, CRF and CG, and (2) a template based probability graph model.

Through analysis and comparison, we are sure that in the social network nodes studied in this paper, the information propagation probability has close relationship with directed edge relationships and the specific propagation approach. We chose the random variable probability based Bayes information propagation network to set up our model.

First, from the traditional undirected clustering coefficient diagram in Figure 1-(a) and 1-(b), we can deduce the edge information propagation probability and direction. In an actual directed social network, the vector influence clustering coefficient should be deduced from Figure 2.

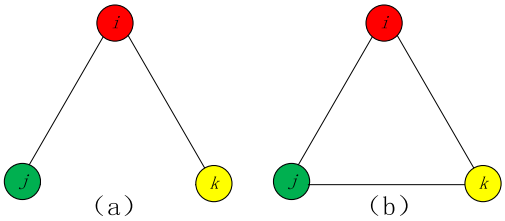


FIGURE 1. Two triple forms of vertex *i* in undirected graph.

We suppose that when all edges in Figure 2-(a) and 2-(b) are bidirectional, they can be equal to Figure 1-(a) and 1-(b) respectively. When all the edges are bidirectional,

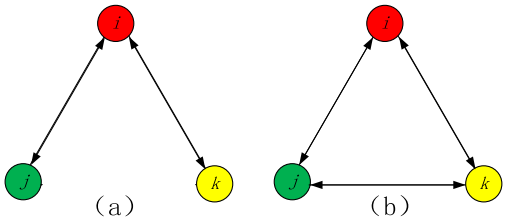


FIGURE 2. Two triple forms of vertex *i* in directed graph.

the information transfer gain path (ITGP) of all the edges is equivalent to that of an undirected network. Tracing back to formula (2), we can split the information contribution (IC) of each edge in the actual existing E_i edges among k_i nodes.

For Figure 2-(a), because there are no edges to connect node j and k, it is a simple situation to analyse. We assume the ITG (information transfer gain) of node i to node j and k, which can be defined in formula (5-1):

$$\mathrm{ITG}_i = \alpha \times \mathrm{ITG}_{i\leftrightarrow j} + \beta \times \mathrm{ITG}_{i\leftrightarrow k} \tag{5}$$

$$\mathrm{ITG}_{i\leftrightarrow j} = \begin{cases} \delta_{i\to j} \times \mathrm{ITG}_{i\to j} + \delta_{i\leftarrow j} \times \mathrm{ITG}_{i\leftarrow j}, & \delta_{i\to j} \neq 0 \\ \theta_{i\to j} \times \delta_{i\leftarrow j} \times \mathrm{ITG}_{i\leftarrow j}, & \delta_{i\to j} = 0 \end{cases} \tag{5-1}$$

$$\mathrm{ITG}_{i\leftrightarrow k} = \begin{cases} \delta_{i\to k} \times \mathrm{ITG}_{i\to k} + \delta_{i\leftarrow k} \times \mathrm{ITG}_{i\leftarrow k}, & \delta_{i\to k} \neq 0 \\ \theta_{i\to k} \times \delta_{i\leftarrow k} \times \mathrm{ITG}_{i\leftarrow k}, & \delta_{i\to k} = 0 \end{cases} \tag{5-2}$$

In the above formula, α is the probability coefficient of $\mathrm{ITG}_{i\leftrightarrow j}$, while β is the probability coefficient of $\mathrm{ITG}_{i\leftrightarrow k}$. ITG_i, $\mathrm{ITG}_{i\leftrightarrow j}$ and $\mathrm{ITG}_{i\leftrightarrow k}$ are the equal bifurcation accumulation parameters for ITG_i. So α and β have the default equal value of 1. $\delta_{i\to j}$, $\delta_{i\leftarrow j}$, $\delta_{i\to k}$ and $\delta_{i\leftarrow k}$ are the information transfer probability parameters of $i \to j$, $i \leftarrow j$, $i \to k$ and $i \leftarrow k$, respectively, given that the default equals a value of 0.5. $\theta_{i\to j}$ and $\theta_{i\to k}$ are the reserve information transfer gain probabilities while calculating the directed information transfer gain clustering coefficient of node i in Figure 2, just like the following relationship in Sina Micro-Blog, Facebook and Twitter. For the calculation of $\mathrm{ITG}_{i\leftrightarrow j}$ and $\mathrm{ITG}_{i\leftrightarrow k}$ in formula (5-1) and (5-2), while there is no edge $i \to j$, $\delta_{i\to j}$'s value is zero which means that j is not a fan of i (i.e., j has no permission to view and share the WeChat moment status of node i). In addition, information cannot be transferred from i to $j(i \to j)$, but there may exist the reserve transferred information from j to i ($i \leftarrow j$), since we are calculating the directed information transfer gain clustering coefficient of vertex i, which is the information transferring source. Compared with the $i \to j$, this direction of $i \leftarrow j$ is a reserve ITG, so we define $\theta_{i\to j} = 0.5$ to represent this case with 50% probability.

For example, in Figure 2-(b), the information transfer gain path (ITGP) is defined as the information starting from node i, going through the path of $i - j$ and $i - k - j$ to give information gain to node j. Based on the probability graph theory model, we can calculate the Bayesian network probability by

abstracting a directed no-loop graph into a Bayesian network. The vertices in the Bayesian network stand for random variables, while edges stand for the probability relation between random variables. Therefore, the joint probability distribution can be represented by the Bayes Chain Rule in formula (6):

$$P(\mathrm{X}_1,\mathrm{X}_2,\ldots,\mathrm{X}_n) = \prod_i P(X_i|Par_G(X_i)) \tag{6}$$

In formula (6), $Par_G(X_i)$ is the corresponding random variable of the parent node of node X_i in Graph G. In the directed network, which we study in this paper, the information influence of a source node on the other nodes brought by the directed transmission of information will also influence the forming of the corresponding community structure. Because of the fact that in real social networks, fans are always gathering around an opinion leader to generate community structure, this phenomenon can influence the information transfer model in the whole network. For Figure 2-(a), we can suppose that node i is the information transferring source for nodes j and k. Based on the probability graph theory, the change of information transferring probability in each edge reflects the information transfer gain received by the end nodes.

According to the above definition of ITGP, we get the formula of ITGP in Figure 2-(b):

$$\begin{aligned}\sum \mathrm{ITG}_{i\leftrightarrow j} &= \mathrm{ITG}_{i\leftrightarrow j} + \mathrm{ITG}_{i\leftrightarrow k\leftrightarrow j} \\ &= \mathrm{ITG}_{i\leftrightarrow j} + \mathrm{ITG}_{i\leftrightarrow k} \times \mathrm{ITG}_{k\leftrightarrow j}\end{aligned} \tag{7}$$

Based on the directivity of ITG and formula (5-1), we can divide $\mathrm{ITG}_{i\leftrightarrow j}$ such that the formula is $\mathrm{ITG}_{i\leftrightarrow j} = \delta_{i\to j} \times \mathrm{ITG}_{i\to j} + \delta_{i\leftarrow j} \times \mathrm{ITG}_{i\leftarrow j}$. Similarly, by symmetry, the sum of $\sum \mathrm{ITG}_{i\leftrightarrow k}$ can be found in formula (8):

$$\begin{aligned}\sum \mathrm{ITG}_{i\leftrightarrow k} &= \mathrm{ITG}_{i\leftrightarrow k} + \mathrm{ITG}_{i\leftrightarrow j\leftrightarrow k} \\ &= \mathrm{ITG}_{i\leftrightarrow k} + \mathrm{ITG}_{i\leftrightarrow j} \times \mathrm{ITG}_{j\leftrightarrow k}\end{aligned} \tag{8}$$

We can also divide $\mathrm{ITG}_{i\leftrightarrow k}$ in formula (8) to $\mathrm{ITG}_{i\leftrightarrow k} = \delta_{i\to k} \times \mathrm{ITG}_{i\to k} + \delta_{i\leftarrow k} \times \mathrm{ITG}_{i\leftarrow k}$. Due to the symmetry of transmission between different nodes, We can suppose the value of $\mathrm{ITG}_{i\to k}$ and $\mathrm{ITG}_{i\leftarrow k}$ is the default unit quantity "1," while the default value of $\delta_{i\to k}$ and $\delta_{i\leftarrow k}$ is 0.5. When there are bidirectional ITG edges between node i and node k, we get $\mathrm{ITG}_{i\leftrightarrow k} = 0.5 \times 1 + 0.5 \times 1 = 1$. By symmetry, we get that while there are bidirectional ITG edges between node i and j, $\mathrm{ITG}_{i\leftrightarrow j} = 0.5 \times 1 + 0.5 \times 1 = 1$.

By setting up the information transfer gain path (ITGP) model of the situation in Figure 2-(a) and Figure 2-(b), we get 9 different ITG sub-figures for Figure 2-(a) in Figure 3 and 27 different ITG sub-figures of Figure 2-(b) (i.e., Figures 4 and 5).

According to formula (5-1), (5-1), (5-2), we calculate the corresponding ITG results in Table 1:

The corresponding ITG_i value of each figure can represent the weight in each type of directed triple. By the statistics of all directed triple types in the graph and the summation by weight, we can calculate all the weight distributions of directed triples in the directed graphs in Table 2 and Table 3.

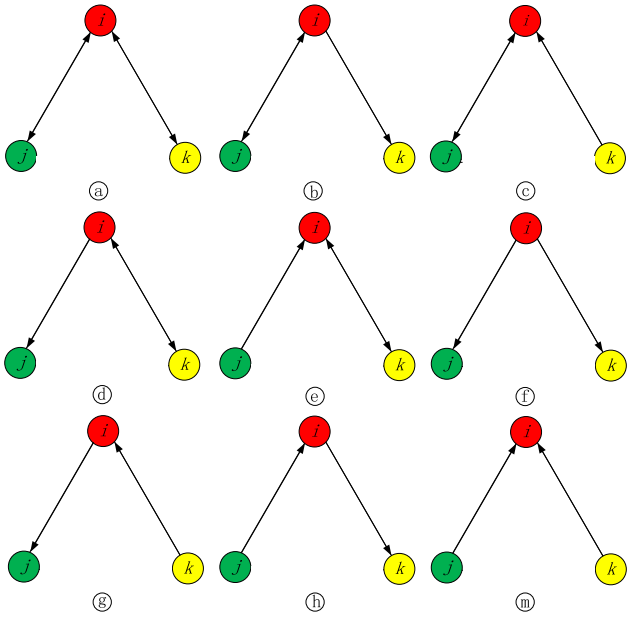


FIGURE 3. Vertex *i* based directed triples (all sub-graphs of Figure 2-a).

TABLE 1. All ITG results of sub-graphs in Figure 3.

Number	Figure Number	ITG_i	$\mathrm{ITG}_{i\leftrightarrow j}$	$\mathrm{ITG}_{i\leftrightarrow k}$
1	Figure 3-a	2	1	1
2	Figure 3-b	1.5	1	0.5
3	Figure 3-c	1.25	1	0.25
4	Figure 3-d	1.5	0.5	1
5	Figure 3-e	1.25	0.25	1
6	Figure 3-f	1	0.5	0.5
7	Figure 3-g	0.75	0.5	0.25
8	Figure 3-h	0.75	0.25	0.5
9	Figure 3-m	0.5	0.25	0.25

TABLE 2. Permutation and combination of 27 conditions.

000	001	002	010	011	012	020	021	022
100	101	102	110	111	112	120	121	122
200	201	202	210	211	212	220	221	222

Similarly, by extending the method to ITG calculation of triangles, we obtain 27 different ITG figures in Figure 4 and Figure 5. Using Figure 4-(a) as an example, node i, node j and node k are all connected with bidirectional edges, while all the edges had been defined as 1 previously. The ITG sum of node i and k consists of single directional ITG of node i and node j and the relayed ITG from k to j starting from i. We see in formula (9) that:

$$\begin{aligned}\sum \mathrm{ITG}_{i\leftrightarrow j} &= \mathrm{ITG}_{i\leftrightarrow j} + \mathrm{ITG}_{i\leftrightarrow k\leftrightarrow j} \\ &= \mathrm{ITG}_{i\leftrightarrow j} + \mathrm{ITG}_{i\leftrightarrow k} + \mathrm{ITG}_{k\leftrightarrow j}\end{aligned} \tag{9}$$

TABLE 3. All ITG results of sub graphs in Figure 4 and Figure 5.

Number	Figure Number	ITG_i	$\sum ITG_{i\leftrightarrow j} = ITG_{i\leftrightarrow j} + ITG_{i\leftrightarrow k\leftrightarrow j} = ITG_{i\leftrightarrow j} + ITG_{i\leftrightarrow k} + ITG_{k\leftrightarrow j}$	$\sum ITG_{i\leftrightarrow k} = ITG_{i\leftrightarrow j} + ITG_{i\leftrightarrow k\leftrightarrow j} = ITG_{i\leftrightarrow k} + ITG_{i\leftrightarrow j} + ITG_{j\leftrightarrow k}$
1	Figure 4-a	3	0.5×1+0.5×(1+1)=1.5	0.5×1+0.5×(1+1)=1.5
2	Figure 4-b	2.375	0.5×1+0.5×(1+(0.5×0.5))=1.125	0.5×1+0.5×(1+0.5)=1.25
3	Figure 4-c	2.375	0.5×1+0.5×(1+0.5)=1.25	0.5×1+0.5×(1+(0.5×0.5))=1.125
4	Figure 4-d	2.5	0.5×1+0.5×(0.5+1)=1.25	0.5×0.5+0.5×(1+1)=1.25
5	Figure 4-e	2.5	0.5×0.5+0.5×(1+1)=1.25	0.5×1+0.5×(0.5+1)=1.25
6	Figure 4-f	1.875	0.5×1+0.5×(0.5+0.5)=1	0.5×0.5+0.5×(1+(0.5×0.5))=0.875
7	Figure 4-g	1.875	0.5×0.5+0.5×(1+(0.5×0.5))=0.875	0.5×1+0.5×(0.5+0.5)=1
8	Figure 4-h	1.875	0.5×1+0.5×(0.5+(0.5×0.5))=0.875	0.5×0.5+0.5×(1+0.5)=1
9	Figure 4-l	1.875	0.5×0.5+0.5×(1+0.5)=1	0.5×1+0.5×(0.5+(0.5×0.5))=0.875
10	Figure 4-m	2.25	0.5×1+0.5×((0.5×0.5)+1)=1.125	0.5×(0.5×0.5)+0.5×(1+1)=1.125
11	Figure 4-o	2.375	0.5×0.5+0.5×(1+1)=1.25	0.5×1+0.5×((0.5×0.5)+1)=1.125
12	Figure 4-p	1.75	0.5×1+0.5×((0.5×0.5)+0.5)=0.875	0.5×0.5+0.5×(1+(0.5×0.5))=0.875
13	Figure 4-q	1.75	0.5×0.5+0.5×(1+(0.5×0.5))=0.875	0.5×1+0.5×((0.5×0.5)+0.5)=0.875
14	Figure 4-r	1.75	0.5×1+0.5×((0.5×0.5)+(0.5×0.5))=0.75	0.5×0.5+0.5×(1+0.5)=1
15	Figure 4-s	1.625	0.5×(0.5×0.5)+0.5×(1+0.5)=0.875	0.5×1+0.5×((0.5×0.5)+(0.5×0.5))=0.75
16	Figure 4-t	2	0.5×0.5+0.5×(0.5+1)=1	0.5×0.5+0.5×(0.5+1)=1
17	Figure 4-u	1.375	0.5×0.5+0.5×(0.5+0.5)=0.75	0.5×0.5+0.5×(0.5+(0.5×0.5))=0.625
18	Figure 4-v	1.375	0.5×0.5+0.5×(0.5+(0.5×0.5))=0.625	0.5×0.5+0.5×(0.5+0.5)=0.75
19	Figure 4-w	1.75	0.5×0.5+0.5×((0.5×0.5)+1)=0.875	0.5×(0.5×0.5)+0.5×(0.5+1)=0.875
20	Figure 4-x	1.75	0.5×(0.5×0.5)+0.5×(0.5+1)=0.875	0.5×0.5+0.5×((0.5×0.5)+1)=0.875
21	Figure 4-y	1.125	0.5×0.5+0.5×((0.5×0.5)+0.5)=0.625	0.5×(0.5×0.5)+0.5×(0.5+0.5×0.5)=0.5
22	Figure 4-z	1.125	0.5×(0.5×0.5)+0.5×(0.5+0.5×0.5)=0.5	0.5×0.5+0.5×((0.5×0.5)+0.5)=0.625
23	Figure 4-α	1.125	0.5×0.5+0.5×((0.5×0.5)+(0.5×0.5))=0.5	0.5×(0.5×0.5)+0.5×(0.5+0.5)=0.625
24	Figure 4-β	1.125	0.5×(0.5×0.5)+0.5×(0.5+0.5)=0.625	0.5×0.5+0.5×((0.5×0.5)+(0.5×0.5))=0.5
25	Figure 4-γ	1.5	0.5×(0.5×0.5)+0.5×((0.5×0.5)+1)=0.75	0.5×(0.5×0.5)+0.5×((0.5×0.5)+1)=0.75
26	Figure 4-ε	0.875	0.5×(0.5×0.5)+0.5×((0.5×0.5)+0.5)=0.5	0.5×(0.5×0.5)+0.5×((0.5×0.5)+(0.5×0.5))=0.375
27	Figure 4-θ	0.875	0.5×(0.5×0.5)+0.5×((0.5×0.5)+(0.5×0.5))=0.375	0.5×(0.5×0.5)+0.5×((0.5×0.5)+0.5)=0.5

Additionally, we obtain the summation of ITG between node i and node k:

$$\sum ITG_{i\leftrightarrow k} = ITG_{i\leftrightarrow k} + ITG_{i\leftrightarrow j} + ITG_{j\leftrightarrow k} \tag{10}$$

Now, ITG of node i is the summation of ITG from node i to the other two nodes. We calculate the ITG of the 27 different figures in Table 2 from Figure 4 and Figure 5:

Because there are three possible directional statuses for each edge in Figures 2-(a) and 2-(b), the adjacent edge of node i has three different definitions, which are friends ($i \leftrightarrow j$), following ($i \rightarrow j$) and fan ($i \leftarrow j$). At the same time, the opposite edge $i \leftrightarrow k$ of node i also has three definitions. Furthermore, the edge $j \leftrightarrow k$ has three types of relationships, which are node j and node k are friends, node j follows node k and node k follows node j. We can use 0, 1, 2 to stand for the relationships and substitute the three different definitions, and we obtain the following 27 arrangements in Table 3, in which the same coloured grid stands for the symmetric figures in Figure 4 and Figure 5:

In the above 27 arrangement cases, because node i is the source node, we can find some symmetry results. For example, Figures 4-(b) and 4-(c) are symmetry results. In Table 3, we use the same colour blocks as the symmetry results, and finally we can get 15 independent results; we separated the sub-graphs into Figure 4 and Figure 5.

The information transfer gain clustering coefficient (ITGC) of node i in a directed network can be summed by the 15 different independent results in Table 3 in formula (11)

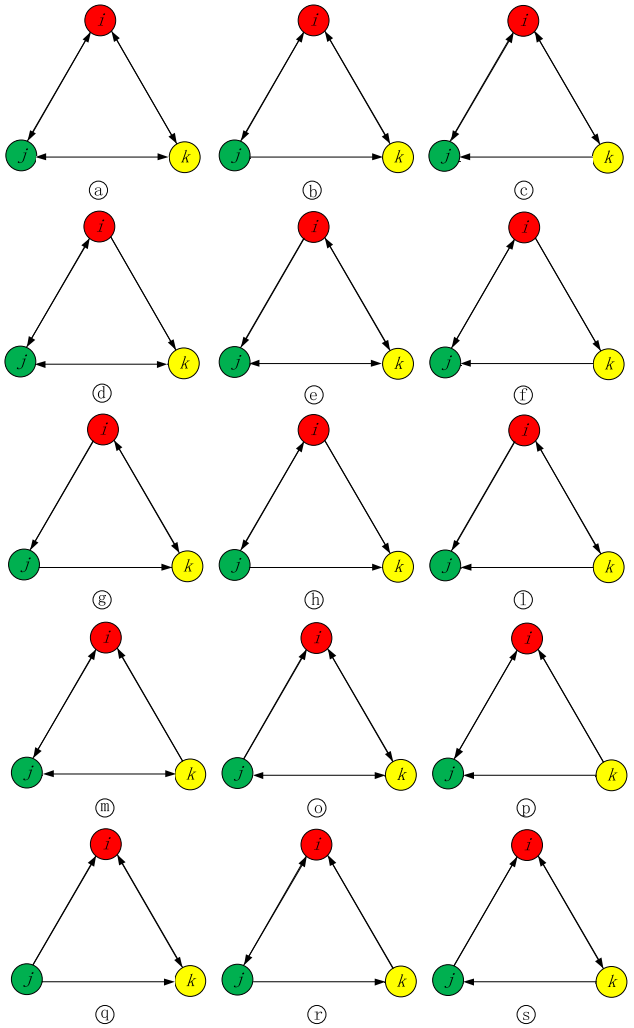


FIGURE 4. Vertex *i* based the first fifteen triangle sub-graphs.

as follows:

$$ITGC_i = \frac{\sum_{t=1}^{15} ITG_{i\_triangle}(t) \times Number(t)}{\sum_{t'=1}^{6} ITG_{i\_triple}(t') \times Number(t')} \tag{11}$$

$ITGC_i$ is the ITG value of node i in a directed network. $\sum_{t=1}^{15} ITG_{i\_triangle}(t) \times Number(t)$ is the weighted number of triangles which use node i as the top vertex (i.e., the information transfer source node), and its weight is the ITG (information transfer gain) contribution $ITG_{i\_triangle}(t)$ from the 15 different types of weighted triangles multiplied by its counted number $Number(t)$. $\sum_{t'=1}^{6} ITG_{i\_triple}(t') \times Number(t')$ is the weighted number of the triples using node i as the top vertex; its weight is the weighted sum of the six $ITG_{i\_triple}(t')$ values of different types of triples multiplied by its counted number $Number(t')$. Similar to undirected clustering coefficients, ITGC has the same characteristic, measuring the tightness of the graph to form tight communities.

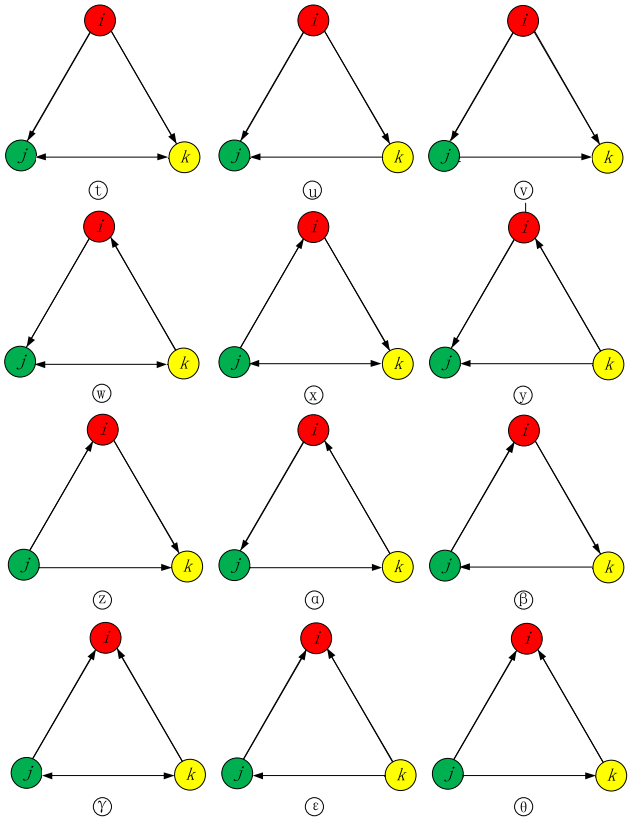


FIGURE 5. Vertex *i* based the last twelve triangle sub-graphs.

B. DIRECTED TARGET FUNCTION CONSTRUCTION AND IMPROVEMENT

We constructed the target function of the vector influence clustering coefficient model. Our model is based on the definition of the weighted community clustering function (WCC) from Prat-Pérez *et al.* [6]. We focused on the properties of the directed graph and made some directional improvement to the model, defining the new target function as Directed Weighted Community Clustering (DWCC). When defining DWCC's relationship of a vertex to its community, we defined $wt(x, C)$ as the weighted triangle number formed by vertex x within community C. In addition, we defined $wvt(x, C)$ as the weighted neighbour node number, which can form triangles by x within community C.

The weighted triangle number $wt(x, C)$ means that based on the focus vertex x, the formed triangle numbers with the definition of ITG can be recorded as a weighted triangle. At the same time, the weighted node number means that the useful and valid node numbers are recorded as weighted nodes in definition of ITG around the focus source vertex x.

Based on optimized iteration of the target function of Arnau Prat-Pérez, the partition refinement step was related to three functions of possible increase of WCC and their similarity, which means that the three different functions can be converted to the united calculation process of WCC_I. In the process of directional improvement, we also need the

estimation of $DWCC_I$ to reduce time complexity of our algorithm, which can be found in the following formula (12):

$$\begin{aligned} DWCC'_I(v, C) &= DWCC(P') - DWCC(P) \\ &= \frac{1}{V} \cdot (d_{in} \cdot \Theta_1 + (r - d_{in}) \cdot \Theta_2 + \Theta_3) \end{aligned} \tag{12}$$

In formula (12) [6], (12-1)–(12-4), as shown at the bottom of this page.

We made some directional improvements to the statistical parameters in formula (12) which can be found in Table 4 because we are handling directed networks.

TABLE 4. Statistical value meaning in $DWCC_I$ estimation.

Parameter	Original Statistic Meaning	Directed Statistic Meaning
r	Node number of community	Node number of community
δ	Edge density of community	Weighted edge density of community
d_{in}	Neighbour number of node in the community	Weighted neighbour number of node in the community
d_{out}	Neighbour number of node out of the community	Weighted neighbour number of node out of the community
b	Boundary edge number of community	Weighted boundary edges number of community
ω	Average clustering coefficient of the whole graph	Average ITG of the whole graph

With the stable values of parameters provided, the calculation of formula (12) has constant time complexity. The updating process of calculated ITG values of each node and $DWCC_I$ only occurs when the structure of a community has been changed. After experiments, the whole time complexity of our algorithm in the entire graph is $O(nm)$, where m is the number of times a community structures changes.

C. ITERATION PROCESS IMPROVEMENT AND ITERATION STOP CONDITION

After finishing the directional improvement above and the construction of $DWCC_I$, we can implement the vector influence based clustering coefficient community detection model by the SCD (Scalable Community Detection) algorithm framework of Arnau Prat-Pérez.

In the initial partition step, we replace the clustering coefficient with the ITG coefficient for the directed graph as the referenced calculation value in the ranking step.

Algorithm 1 Partition Refinement

Input: Graph $G(V,E)$ and Partition P
Output: newPartition P'

```
newP = P
newDWCC = computeDWCC(P)
maxDWCC = newDWCC
preMaxDWCC = 0
flag = FALSE
counter = 0
while NOT(flag AND (maxDWCC − preMaxDWCC)/preMaxDWCC < t) OR counter < 20 do
    flag = FALSE
    DWCC' = newDWCC
    P' = newP
    M = ∅
    for each v in V do
        M.add(BestMovement(v,P'))
    end for
    newP=applyMovements(M,P')
    newDWCC=computeWCC(newP)
    if maxDWCC < newDWCC then
        flag = TRUE
        preMaxDWCC = maxDWCC
        count = 0
        maxDWCC = newDWCC
    end if
    count++
end while
return P'
```

FIGURE 6. Algorithm framework of partition refinement.

In the partition refinement step, we found that the greed of the original algorithm cannot be ignored where the structure of the original algorithm cannot converge efficiently. We improve the structure of iteration in our algorithm by adding the flag bit *flag* to label the current iteration, which has reached the maximum $DWCC$ updating limit value or not.

By obtaining the maximum value of $DWCC$, we simulate the progress to reach the local optimum solution, which can be found in Figure 6. According to the experimental process, we have set up the iteration stop condition to meet one of the following two conditions:

- There are still updates of maximum $DWCC_I$ but the update proportion of $DWCC_I$ value is no less than the threshold t;
- There is no update of maximum $DWCC_I$ and the iteration has been processed 20 times.

$$\Theta_1 = \frac{((r-1)\delta + 1 + q)(d_{in} - 1)\delta}{(r+q)((r-1)(r-2)\delta^3 + (d_{in}-1)\delta + q(q-1)\delta\omega + q(q+1)\omega + d_{out}\omega)} \tag{12-1}$$

$$\Theta_2 = -\frac{(r-1)(r-2)\delta^3}{(r-1)(r-2)\delta^3 + q(q-1)\omega + q(r-1)\delta\omega} \cdot \frac{(r+1)\delta + q}{(r+q)(r-1+q)} \tag{12-2}$$

$$\Theta_3 = \frac{d_{in}(d_{in}-1)\delta}{d_{in}(d_{in}-1)\delta + d_{out}(d_{out}-1)\omega + d_{out}d_{in}\omega} \cdot \frac{d_{in} + d_{out}}{r + d_{out}} \tag{12-3}$$

$$q = (b - d_{in})/r \tag{12-4}$$

TABLE 5. Dataset attributes.

Dataset	Node	Edge	Edge density
Artificial dataset	30	275	9.1667
OSLOM	301	6234	20.7110
Subject reference	40	306	7.6500
Calling record	284	3934	13.8521

To our surprise, it was proven by the experimental process that our iteration stop conditions can remove the algorithm running situation of ''trapping in local optimum solution'' to a certain extent and that these conditions do not consume much execution time or increase the time of iterations.

IV. DATASET AND RESULT ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

The CPU frequency of Master node is Intel(R) Xeon(R) CPU E5-2440 v2 @1.90 GHz. Memory is 16 GB with 4 TB hard disk. JDK version is 1.8.0 131.

B. EXPERIMENT RESULTS

Four representative datasets were selected to evaluate our algorithm: an artificial simulation dataset, two classic datasets in community detection and a real world dataset for connected closely crowd calling record in a city of China.

Dataset 1: an artificial dataset generated by three communities randomly. We generated it by forming an edge in the community with probability 0.5 and forming an edge between communities with probability 0.25.

Dataset 2: an OSLOM dataset, provided by opening the classical source algorithm OSLOM as an example dataset [17].

Dataset 3: a subject reference dataset, provided by the Infomap algorithm as a subject reference dataset [13].

Dataset 4: a real-world mobile calling dataset from one Chinese city's cell phone calling records in one month (see Table 5).

In this article, we compare the performance of our ITG-DWCC algorithm with the following classical directed community detection algorithms:

- FastGN: provided by Newman, which is the classical fast community detection algorithm [3]
- OSLOM: provided by Lancichinetti *et al.* [17]
- Infomap: provided by Rosvall and Bergstrom [13].

We use the following comparison indicators to analyse different algorithms that are often adopted by some authoritative researchers [3] in this area:

- Community Number: the number of communities after community detection,
- Directed Modularity: the calculation of a directed graph's modularity [20],

$$Q = \frac{1}{m}\sum_{i,j}[A_{ij} - \frac{k_i^{out}k_j^{in}}{m}] \cdot \delta(c_i, c_j) \qquad (13)$$

- Jaccard: the calculation of result precision. Its definition formula uses authoritative result and the intersection of algorithm result of set over union of set [21],

TABLE 6. Community number.

Dataset	FastGN	OSLOM	Infomap	ITG-DWCC
Artificial dataset	5	Low quality	1	**4**
OSLOM	9	9	3	**10**
Subject reference	4	4	4	**3**
Calling record	8	14	22	**18**

TABLE 7. Directed modularity.

Dataset	FastGN	OSLOM	Infomap	ITG-DWCC
Artificial dataset	0.3322	Low quality	1.0	**0.3588**
OSLOM	0.8651	0.8794	0.9963	**0.6910**
Subject reference	0.8007	0.8072	0.8007	**0.8235**
Calling record	0.5486	0.4822	0.3915	**0.4080**

TABLE 8. Jaccard.

Dataset	FastGN	OSLOM	Infomap	ITG-DWCC
Artificial dataset	0.2448	Low quality	0.3333	**0.2660**
OSLOM	0.9592	1.0	0.3851	**0.5183**
Subject reference	1.0	0.9070	0.9310	**0.5974**
Calling record	1.0	0.2864	0.2448	**0.1121**

TABLE 9. F-1.

Dataset	FastGN	OSLOM	Infomap	ITG-DWCC
Artificial dataset	0.3830	Low quality	0.5	**0.4075**
OSLOM	0.9700	1.0	0.3851	**0.6386**
Subject reference	1.0	0.9070	0.9536	**0.7108**
Calling record	1.0	0.4149	0.3611	**0.1945**

- F-measure: the measure of the result of the algorithm; in our experiment, we use F-1 standard [21].

To avoid random deviations, we executed the following algorithms 10 times each and obtained the experimental results presented in Tables 6–9:

The experimental results indicate that for the above datasets, compared with other algorithms, the ITG-DWCC algorithm reached precision in an accepted scale. In some cases, the ITG-DWCC algorithm performed better than traditional community detection algorithms. In Table 6, we find that for the OSLOM and Calling record datasets, the Community Number of ITG-DWCC is more accurate than that of the FastGN and OSLOM algorithms. In addition, in Table 7, for the Subject reference dataset, ITG-DWCC has much better Directed Modularity values (see Table 8).

In terms of scalability, because the step of calculating the best movement can be deployed in a distributed environment, our ITG-DWCC algorithm has an advantage for large-scale networks of parallel computing, which can be used for efficient community detection on large-scale directed networks. To test the parallel performance of our Distributed ITG algorithm (DITG-DWCC), we used some real directed large calling record networks from a calling graph in a city of China; it can be found in Table 10. The experimental results of DITG compared to other classical directed community detection algorithms are presented in Table 11.

For the experiment whose results are presented in Table 11, we used the following parallel computing environment to test the performance of DITG-DWCC: eight slave nodes were used in deployment with Spark version 2.1.0. and Hadoop

TABLE 10. Large directed network datasets.

Dataset	Vertex	Edges	Edges density
Call L-1	13,310	34,591	0.3847
Call L-2	29,624	55,423	0.5345
Call L-3	61,510	65,202	0.9433
Call L-4	512,024	1,021,861	0.5011

TABLE 11. Time consumed using 8 nodes (seconds).

Dataset	FastGN	OSLOM	Infomap	DITG-DWCC
Call L-1	2906.124	3002.762	3230.453	**950.125**
Call L-2	2867.634	2994.986	2898.872	**901.651**
Call L-3	4676.767	5877.877	6030.331	**1500.765**
Call L-4	46778.771	49001.225	50020.222	**8228.472**

version 2.7.3 on each node. The CPU of Slave nodes were Intel(R) Xeon(R) CPU E5-2440 v2@1.90 GHz. Their memories are 16 GB each.

Table 10 shows that with the scale growth of the dataset, our DITG algorithm has very good distributed performance in handling large-scale directed networks, better than FastGN, OSLOM and Infomap in terms of time consumed.

V. CONCLUSIONS

First, this paper put a classic probability graph and clustering coefficient together and proposed a new vector influence-based clustering coefficient ITG for measuring directed graphs. Then, we combined the definition of target function iterative optimization of DWCC with directed modularity in the community detection of directed networks. In the iterations, due to the independent optimal movement calculation, we can perform a paralleling operation in the most time-consuming step.

With extensive experiments in artificial network datasets and real-world, large network datasets derived from online social media, it has been proved that our algorithm is more accurate and faster than several traditional and well known community detection methods such as FastGN, OSLOM and Infomap in directed networks. Our ITG-DWCC algorithm has acceptable precision and has obvious advantages regarding time complexity.

Our follow-up research will include: (1) optimizing the stop condition in the iterations to fetch up the greediness of the algorithm, (2) implementing efficiency of our parallel algorithm and experimenting in a distributed environment, and (3) integrating our algorithm to form a visualization analysis software application.

ACKNOWLEDGEMENT

The authors appreciate direction from professor Hui Zhang and his aid from the Joint-Operated project from the National Natural Science Foundation of China (NSFC) from Tsinghua University.

REFERENCES

[1] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surveys*, vol. 45, no. 4, p. 43, 2013, doi: 10.1145/2501654.2501657.

[2] M. E. J. Newman and M. Girvan, "Community detection in networks: Modularity optimization and maximum likelihood are equivalent," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 94, p. 052315, Jun. 2016, doi: 10.1103/PhysRevE.94.052315.

[3] 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, doi: 10.1103/PhysRevE.69.026113.

[4] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, p. 066111, Dec. 2004, doi: 10.1103/PhysRevE.70.066111.

[5] V. D. Blondel, J. L. Guillaume, and R. Lambiotte, "Fast unfolding of communities in large networks," *J. Statist. Mech., Theory Experim.*, vol. 10, p. P10008, Oct. 2008, doi: 10.1088/1742-5468/2008/10/P10008.

[6] A. Prat-Pérez, D. Dominguez-Sal, and J. L. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 225–236, doi: 10.1145/2566486.2568010.

[7] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 912–919.

[8] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, p. 036106, 2007, doi: 10.1103/PhysRevE.76.036106.

[9] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, p. 103018, 2010, doi: 10.1088/1367-2630/12/10/103018.

[10] D. Jin, D. Y. Liu, and B. Yang, "Fast complex network clustering algorithm using local detection," (in Chinese), *DianziXuebao (Acta Electron. Sinica)*, vol. 39, no. 11, pp. 2540–2546, 2011.

[11] G. Cordasco and L. Gargano, "Community detection via semi-synchronous label propagation algorithms," in *Proc. IEEE Int. Workshop Bus. Appl. Social Netw. Anal. (BASNA)*, Bangalore, India, Dec. 2010, pp. 1–8, doi: 10.1109/BASNA.2010.5730298.

[12] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proc. Int. Symp. Comput. Inf. Sci.*, 2005, pp. 284–293, doi: 10.1007/11569596_31.

[13] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, 2008, doi: 10.1073/pnas.0706851105.

[14] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, p. 056117, 2009, doi: 10.1103/PhysRevE.80.056117.

[15] G. Palla, I. Derényi, and I. Farkas, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005, doi: 10.1038/nature03607.

[16] Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010, doi: 10.1038/nature09182.

[17] A. Lancichinetti, F. Radicchi, and J. Ramasco, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, p. e18961, 2011, doi: 10.1371/journal.pone.0018961.

[18] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, Rome, Italy, 2013, pp. 587–596, doi: 10.1145/2433396.2433471.

[19] M. E. J. Newman and A. Clauset, "Structure and inference in annotated networks," *Nature Commun.*, vol. 7, p. 11863, Jun. 2016, doi: 10.1038/ncomms11863.

[20] X. Zhang, T. Martin, and M. E. J. Newman, "Identification of core-periphery structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, p. 032803, Mar. 2015, doi: 10.1103/PhysRevE.91.032803.

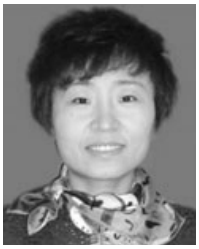
[21] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA, MIT Press, 2009.

[22] V. Levorato and C. Petermann, "Detection of communities in directed networks based on strongly p-connected components," in *Proc. Int. Conf. Comput. Aspects Social Netw. (CASoN)*, 2011, pp. 211–216, doi: 10.1109/CASON.2011.6085946.
[23] A. Arenas, J. Duch, and A. Fernández, "Size reduction of complex networks preserving modularity," *New J. Phys.*, vol. 9, no. 6, p. 176, 2007, doi: 10.1088/1367-2630/9/6/176.
[24] J. Leskovec and R. Sosic, "SNAP: A general-purpose network analysis and graph-mining library," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, pp. 1–20, 2016, doi: 10.1145/2898361.
[25] A. Grover and J. Leskovec, "NODE2VEC: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864, doi: 10.1145/2939672.2939754.
[26] X. Deng, W. Ying, and Y. Chen, "Highly efficient epidemic spreading model based LPA threshold community detection method," *Neurocomputing*, vol. 210, pp. 3–12, Oct. 2016, doi: 10.1016/j.neucom.2015.10.142.
[27] P. G. Sun and L. Gao, "A framework of mapping undirected to directed graphs for community detection," *Inf. Sci.*, vol. 298, pp. 330–343, Mar. 2015, doi: 10.1016/j.ins.2014.10.069.
[28] X. L. Deng and Y. X. Li, "MapReduce-based efficient betweenness approximation pivot method for large graphs," *Int. J. Inf. Technol. Manage.*, vol. 15, no. 2, p. 144, 2016, doi: 10.1504/IJITM.2016.076394.
[29] J. Liu, C. Aggarwal, and J. Han, "On integrating network and community discovery," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 117–126, doi: 10.1145/2684822.2685323.
[30] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 531–549, 1st Quart., 2017.
[31] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 225–235, Jan. 2017.
[32] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.
[33] S. Yu, G. Wang, and W. Zhou, "Modeling malicious activities in cyber space," *IEEE Netw.* vol. 29, no. 6, pp. 83–87, Jun. 2015.

XIAOLONG DENG was born in 1977. He received the Ph.D. degree. He is currently an Assistant Professor and a Master Supervisor in data mining with the Beijing University of Posts and Telecommunications. His research interests include data mining and complex networks.

JIAYU ZHAI was born in 1995. He received the bachelor's degree. He is currently a Senior Student with the Beijing University of Posts and Telecommunications and the Queen Mary University of London. His research interests include social network analysis and data mining.

TIEJUN LV (SM'12) was born in 1969. He received the Ph.D. degree. He is currently a Professor with the Beijing University of Posts and Telecommunications. His research interests include network communication and signal processing.



LUANYU YIN was born in 1974. She received the Ph.D. degree. She is currently a Professor with the China Academy of Social Management, Beijing Normal University. Her research interests include public service and society governance.