IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 30, 2020, accepted February 26, 2020, date of publication March 2, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977824

NALPA: A Node Ability Based Label Propagation Algorithm for Community Detection

YUN ZHANG[1], YONGGUO LIU[1], JIAJING ZHU[1], CHANGHONG YANG[2], WEN YANG[2], AND SHUANGQING ZHAI[3]

[1]Knowledge and Data Engineering Laboratory of Chinese Medicine, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Sichuan Center for Disease Control and Prevention, Chengdu 610041, China
[3]School of Traditional Chinese Medicine, Beijing University of Chinese Medicine, Beijing 100029, China

Corresponding author: Yongguo Liu (liuyg@uestc.edu.cn)

This work was supported in part by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant 2018ZX10715003-002, in part by the National Key Research and Development Program of China under Grant 2017YFC1703905 and Grant 2018YFC1704105, and in part by the Sichuan Science and Technology Program under Grant 2018GZ0192, Grant 2019YFS0019 and Grant 2019YFS0283.

ABSTRACT Community is an important topological characteristic of complex networks, which is significant for understanding the structural feature and organizational function of networks, and community detection has recently attracted considerable research effort. Among community detection methods, label propagation technology is widely used because of its linear time complexity. However, due to the randomness of the node order of label updating and the order of label launching in label propagation, the instability of community detection approaches based on label propagation becomes a challenge. In this paper, a new label propagation algorithm, Node Ability based Label Propagation Algorithm (NALPA), is proposed to discover communities in networks. Inspired from human society and radar transmission, we design four node abilities (propagation ability, attraction ability, launch ability and acceptance ability), label influence and a new label propagation mechanism to deal with the instability and enhance the efficiency. Experimental results on 42 synthetic and 14 real-world networks demonstrate that NALPA outperforms state-of-the-art approaches in most cases. In a case study, NALPA is applied to a drug network in Traditional Chinese Medicine (TCM) and can discover the drug communities where drugs have similar efficacy for treating Chronic GlomeruloNephritis (CGN).

INDEX TERMS Community detection, label influence, label propagation, node ability.

I. INTRODUCTION

Complex systems can be viewed as complex networks, such as social networks, traffic networks, biological networks, etc., where nodes denote objects and edges indicate the interaction among objects [1]. Analyzing the structural feature and organizational function of complex networks is an important research area. Among different features of networks, community has received widespread attention, which is the division of a network into the groups of nodes having dense intra-connections and sparse inter-connections [2]. Overlapping nodes are shared among different communities in networks [3]. Meanwhile, each node has different importance reflecting its weight itself, for example, a node with large degree or locating in the center of networks owns large importance [4], [5].

Significant researches have been carried out in identifying communities, which is helpful for understanding and exploiting networks more effectively [6], [7]. For example, detecting communities in citation networks might find the papers on related topics [8]. Community structures of complex networks can be revealed by community detection algorithms [9], [10], including modularity optimization, spectral clustering, hierarchical partition, label propagation, information theory based algorithms, and so on [11], [12]. In this paper, we pay attention to label propagation based algorithms for community detection because of their simple idea and near-linear time complexity. Label Propagation Algorithm (LPA) is a fast community detection algorithm with nearly linear complexity [12]. But it is known that

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

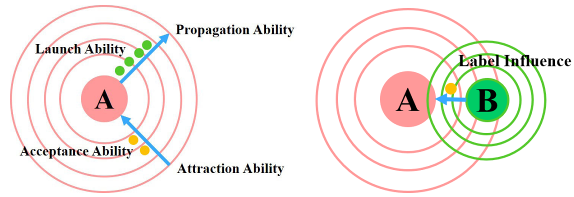
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

LPA cannot detect overlapping communities in networks and provide stable community partition, since a node only owns one label and the node order of label updating is random. Community Overlap PRopagation Algorithm (COPRA) is a modified LPA method to uncover overlapping communities by assigning nodes multiple labels with belonging coefficients [13]. If the belonging coefficients of labels are smaller than $1/\gamma$, the labels are filtered, where γ is a threshold to regulate the number of communities that nodes can belong to. When COPRA stops, if a node has several labels, in other words, it is an overlapping node, then it is allocated to multiple communities. However, COPRA still cannot obtain stable communities, since the node order of label updating is also random and COPRA randomly chooses one label in multiple labels with same belonging coefficient of a node to propagate to other nodes. To better discover communities, researchers proposed many label propagation algorithms to handle the instability problem by considering node and label importance (their weight) to decide the updating order [7], [14]–[19].

By analyzing the label propagation process in above methods, we find that there are some limits: 1) the node order of label updating is random, 2) node importance only measures the weight of nodes themselves but does not evaluate their influence to others, 3) a node only accepts the labels from its neighbor nodes and spreads one label to others once although this node may own many labels, 4) the weight of labels is constant when the labels are launched from a node to another node, which may cause the instability problem and lead methods detecting erroneous communities. Some researches [7], [14], [17]–[20] are proposed to solve the first problem to enhance the stability. To our best knowledge, there are few researches to deal with 2) - 4) problems.

In this paper, a new label propagation algorithm, Node Ability based Label Propagation Algorithm (NALPA), is proposed for community detection in networks. Inspired from real world, we design four node abilities (propagation ability, attraction ability, launch ability and acceptance ability) and label influence to provide a new label propagation mechanism and confirm the node order of label updating and the order of label launching and accepting for enhancing efficiency and improving the instability.

In society, people have large importance if they own a senior role, then they have strong capability to propagate more views to others in wider range, and vice versa [21], [22]. On the other hand, the influence of a person also comes from its friends [21], [22]. For example, under the condition that people A and B own the same importance themselves, if the friends of person A have stronger importance than those of person B, then person A is often considered to own larger influence than person B. Thus, we can consider that the total influence of a person consists of personal importance and friends' importance. Then they are affected by the ones with small influence in low possibility [21], [22]. By modeling people as nodes and their relations as edges, we can construct a social network to describe people and their relations. Like human society, a node has large importance



(a) four abilities (b) label influence

FIGURE 1. Illustration of four abilities and label influence.

itself if it has large degree or locates at the central position of networks [4], [5]. The node with larger influence has stronger ability to receive the labels from farther nodes and spread more labels to farther nodes [21], [22]. Four ability factors of nodes, propagation ability, attraction ability, launch ability and acceptance ability, are defined to model the influence of people as shown in Figure 1(a). **Propagation ability and attraction ability** can reflect the influence range of a node, which consist of personal importance and neighbor importance. Propagation ability denotes the distance where the labels of nodes can be spread. Attraction ability denotes the distance where nodes can receive the labels from other nodes. The other two factors, **launch ability and acceptance ability**, reflect the number of labels that a node can spread and accept, respectively. In order to handle the randomness, we fix the node order of label updating in the ascending order of propagation ability. Different with node importance [7], [15]–[19], the four abilities of nodes can reflect the influence of nodes to others, which include the influence range and the quantity of label launching and accepting. A node can receive labels from farther nodes according to propagation ability and attraction ability and can launch and accept multiple labels according to launch ability and acceptance ability, which is beneficial for detecting accurate communities and converging fast.

The second inspiration is derived from radar transmission. If two radars are in the coverage of each other, they identify each other and begin to transfer information. The transmission intensity depends on radar power, the distances between radars and the importance of information [23], [24]. After defining the ability factors of nodes, we design a new label propagation mechanism and label influence to mimic the mode and intensity of radar transmission as shown in Figure 1(b), which is helpful for accelerating the label propagation among nodes and discovering accurate communities. Like radar transmission, we design **the label propagation mechanism** including the steps 1) determining the nodes which can launch labels to the updating node, 2) determining the labels of nodes which are launched and their quantity, 3) determining the labels which are accepted by the updating node and their quantity and 4) starting propagation. We find that existing label propagation based algorithms consider that the weight of labels of a node (i.e., belonging coefficients) is unchanged when labels arrive at other nodes [7], [14], [17]–[20]. Similar with radar transmission,

TABLE 1. Corresponding concepts.

Real world	NALPA
influence range	propagation and attraction ability
human propagate views	launch ability
human is affected by others	acceptance ability
radar transmission	label propagation mechanism
transmission intensity	label influence

the weight of labels should change according to the importance of launcher, the distances among nodes and the belonging coefficients of labels. Thus, we define **label influence** to reflect label weight when they arrive at a node by the above three factors in NALPA. As a result, NALPA filters the labels with small influence to improve accuracy and accelerate convergence. If a label is more influential than others, then nodes are assigned to the community characterized by this label more possibly. The corresponding concepts between real world and NALPA are listed in Table 1.

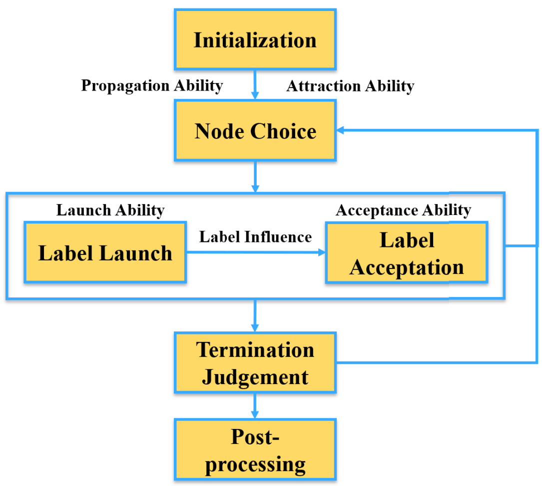


FIGURE 2. The framework of NALPA.

The framework of NALPA is shown in Figure 2. Firstly, all nodes are initialized with labels, such as node id. Secondly, node v_i is chosen to update its labels according to propagation ability. Then the surrounding nodes that can launch labels to updating node v_i are decided by propagation and attraction ability. Thirdly, surrounding node v_j of node v_i chooses labels owning large belonging coefficient to spread to node v_i until the number of propagating labels equals to the launch ability of node v_j. During label propagation process, labels are assigned with label influence when it arrives at node v_i. Fourthly, node v_i accepts the labels with large label influence until the number of accepting labels equals to the acceptance ability of node v_i. Above steps except initialization are executed iteratively until all nodes are updated. Finally, if NALPA reaches termination condition, nodes are assigned to communities by post-processing, else NALPA begins next iteration.

Experimental results on synthetic and real-world networks show that our proposed method provides better results than state-of-the-art algorithms in most networks. The main contributions can be summarized as follows.

- Four node abilities are defined to describe the influence of nodes to provide a new label propagation mechanism and decided the node order of label updating.
- Label influence is defined to reflect the importance of labels when the label arrives at other nodes and determine the order of label accepting.
- We evaluate the effectiveness of NALPA on 42 synthetic and 14 real-world networks.

The rest of the paper is organized as follows. Section II reviews the related work. Section III defines several terms associated with node abilities and label influence. Section IV describes our proposed method NALPA in detail. The experimental setups and results are given in Sections V and VI, respectively. Finally, Section VII concludes the paper.

II. RELATED WORK

A considerable amount of literature has been published on finding communities in networks. We highlight a few ideas that use label propagation to discover communities. Raghavan *et al.* [12] introduced label propagation into community detection. Each node updates its label to another one which is the most popular among its neighbors in LPA. When LPA stops, the nodes with the same label are assigned to the same community. However, LPA only can detect non-overlapping communities in networks, and the discovered communities have randomness because LPA randomly chooses nodes to update. COPRA [13] and SLPA [25] are proposed to discover overlapping communities in networks. In COPRA, each node owns a set of labels, and labels have the belonging coefficients reflecting their membership among communities, which are updated by averaging the belonging coefficients over all labels in neighbors. SLPA is proposed for overlapping community detection based on the dynamic process of the speaker-listener interaction. Nodes have the memory spaces to store received labels, then nodes are allocated to communities according to the frequency of labels in the memory spaces when iterations end.

To better discover communities, researchers proposed many label propagation algorithms to handle the instability problem by considering the weight of nodes and labels. Tong *et al.* [14] proposed Weighted Label Propagation Algorithm (WLPA) to consider the weight of labels, which is defined by the ratio of nodes' degree to the average degree of the networks. WLPA spreads the label with the largest weight to handle the randomness. Sun *et al.* [20] proposed Dominant Label Propagation Algorithm (DLPA) simulating the voting process to consider the weight of neighbors. The confidence of neighbors to a node is defined to measure the importance of neighbors and is used to calculate dominant labels. As a result, DLPA propagates dominant labels to improve the instability. However, DLPA also updates the labels of nodes in a random order, which leads to the instability concern.

Liu *et al.* [15] proposed DLPA$^+$ and introduced the confidence variance of nodes to improve DLPA's instability by updating the labels of nodes according to their confidence variance.

Apart from considering the neighbors of a node, researchers also focused on the node itself. Xing *et al.* [16] proposed a Node Importance Based Label Propagation Algorithm (NIBLPA) to deal with the instability of LPA. In NIBLPA, node importance is defined by k-shell decomposition to measure the weight of nodes and decide the node order of label updating, and the labels of more important nodes are updated earlier. In addition, label importance is defined by the node importance and degree of nodes to determine the order of label choosing. Nodes choose the label with the largest label importance to propagate. Based on NIBLPA, Wu *et al.* [17] proposed LINSIA, in which node and label importance are also defined to deal with the instability. Different with NIBLPA, LINSIA utilizes Extended Neighborhood Coreness centrality (ENCoreness) to replace k-shell decomposition values and define the node importance, since the k-shell decomposition fails to yield the monotonic influence ranking of nodes because it assigns many nodes with the same k-shell value. Meanwhile, based on the label importance defined in NIBLPA, LINSIA calculates the label importance by multiplying an extra ratio, which denotes the membership of node v_j belonging to the community characterized by label l. Zhang *et al.* [18] proposed label propagation algorithm LPA_NI based on node and label importance. Different with NIBLPA and LINSIA, LPA_NI defines the node importance by considering the priori importance of nodes calculating by Bayesian network from expert knowledge. LPA_NI determines the node order of label updating by node importance and selects important labels to update the labels of nodes. Berahmand and Bouyer [9] designed a label propagation algorithm LP-LPA. Based on semi-local similarity methods, link strength is defined to measure the relation power between two nodes and label influence is considered to measure the popularity of each community label for selecting initial nodes and avoiding instability. Shen and Ma [19] proposed a Node Gravitation based Label Propagation Algorithm (NGLPA), in which labels are propagated by node gravitation defined by node importance and node similarity. Different with above methods, NGLPA defines node importance by LeaderRank [26] that computes the relations among nodes and its neighbors iteratively and defines node similarity by Jaccard index.

Some researchers also considered nodes and their neighbors simultaneously. Lu *et al.* [7] proposed a label propagation algorithm LPANNI based on node importance, node similarity and neighbor node influence, in which node importance is defined by node degree and the number of the triangles formed by the node and its neighbors. Node similarity is defined by a variation of Jaccard index, in which the direct or indirect paths between nodes v_i and v_j are considered and path length threshold α denoting the maximum of paths is used to control computing complexity. Neighbor node influence is defined by node importance and node similarity to measure the influence of neighbor nodes on node v_i. LPANNI updates nodes in the ascending order of node importance and calculates belonging coefficients based on the neighbor node influence.

Despite their initial success, there are still limitations for existing approaches as mentioned above. To lift the limitations, in this paper, we provide some ways as follows.

- Four node abilities are defined to measure fine-grained node influence including the influence range and the quantity of label launching and accepting to form a new label propagation mechanism, which may be beneficial for discovering accurate communities and converging rapidly.
- Nodes can accept labels from farther nodes besides neighbor nodes and launch multiple labels to other nodes once, which can use more node and label information and help the model detect communities fast and accurately.
- Label influence is defined to measure the weight of labels, which will change when they arrive at other nodes during label propagation to help the model to detect accurate membership of nodes in communities.

III. PRELIMINARIES

Given unweighted and undirected network $G = (V, E)$, where $V = \{v_1, \ldots, v_i, \ldots, v_N\}$ represents node set, $E = \{e_1, \ldots, e_j, \ldots, e_M\}$ represents edge set, and N and M denote the number of nodes and edges, respectively. The neighbor set of node v_i is expressed as $N(v_i) = \{v_j | e_{v_i v_j} \in E\}$, and k_{v_i} denotes the degree of node v_i. The label set of node v_i is expressed as $B(v_i) = \{(l_1^{v_i}, c_1^{v_i}), \ldots, (l_j^{v_i}, c_j^{v_i}), \ldots, (l_H^{v_i}, c_H^{v_i})\}$, where label $l_j^{v_i}$ with belonging coefficient $c_j^{v_i}$ is the jth label of node, $\sum_{j=1}^{H} c_j^{v_i} = 1$, and H is the number of labels.

A. PROPAGATION ABILITY

The propagation ability of node v_i reflects its influence range and denotes the distance where its labels can be spread, which is defined as

$$P_{v_i} = C_{v_i} \times k_{v_i} + \alpha \sum_{v_j \in N(v_i)} \frac{k_{v_j}}{\sum_{v_k \in N(v_i)} k_{v_k}} \times C_{v_j} \times k_{v_j}, \tag{1}$$

where $C_{v_i} \times k_{v_i}$ is the personal importance of node v_i, which is defined by closeness centrality C_{v_i} and degree k_{v_i} [4], [5]. The second item represents the neighbor importance of node v_i, which is influenced by three factors, the weight, closeness centrality and degree of neighbor v_j. Closeness centrality C_{v_i} of node v_i evaluates its centrality in networks, which is defined as

$$C_{v_i} = \frac{N-1}{\sum_{v_j \in V} d_{v_i v_j}}, \tag{2}$$

where $d_{v_i v_j}$ is the shortest distance between nodes v_i and v_j. Influence factor $\alpha \in [0, 1]$ is used to adjust neighbor importance to balance the personal and neighbor importance of node v_i. For handling the computing complexity of Eq. (2), we set $v_j \in N(v_i)$ in large networks. In order to deal with the instability problem, we update the labels of nodes in the ascending order of propagation ability, since the nodes with small propagation ability can be affected by other nodes easily.

B. ATTRACTION ABILITY

The attraction ability of node v_i also reflects its influence scope, however, it is different with propagation ability and denotes the distance where it receives labels from other nodes, which is defined as

$$A_{v_i} = |P_{v_i}|. \tag{3}$$

In society, if people can spread information in certain range, they can obtain information in the same range [21].

C. LAUNCH ABILITY

Most label propagation algorithms only send a label once in spreading process [7], [12]–[19], [25]. Spreading all labels of node v_i cannot obtain satisfying results [13]. However, the labels of node v_i contain node and community information [18]. If the methods only spread one label in the propagation process, they need more iterations for discovering communities because of losing some node and label information possibly. Thus, we propagate partial labels according to the launch ability of node v_i to catch more network information. The launch ability of node v_i represents the number of labels that node v_i can propagate to other nodes, which is defined as

$$L_{v_i} = \lfloor P_{v_i}^{\beta} \rfloor. \tag{4}$$

Launch factor $\beta \in [0, 1]$ is used to adjust the number of labels sent by node v_i. If the launch ability of node v_i is smaller than 1, we make $L_{v_i} = 1$ to ensure that node v_i can spread a label. In addition, node v_i chooses the labels with large belonging coefficients to launch. In real world, if people have large influence, they can send plenty of information to others [21], [22].

D. ACCEPTANCE ABILITY

In the procedure of label propagation, many labels with different label influence may arrive at node v_i. The label influence of some labels is too small to have impact on community partition. In this paper, we define the acceptance ability of node v_i as the number of labels that a node accepts, which is defined as

$$R_{v_i} = \gamma. \tag{5}$$

Here, the acceptance ability is defined as a global threshold $\gamma \in N^+$ for simply filtering labels with small weight and controlling the number of communities that nodes can belong to [13]. Node v_i accepts labels until their quantity equals to its acceptance ability. In real world, people only accept information which is necessary and important for them [21], [22].

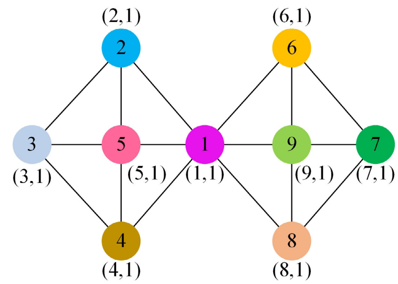


FIGURE 3. A sample network.

TABLE 2. Node abilities of the sample network.

Node ID	1	2	3	4	5	6	7	8	9
C_{v_i}	0.80	0.57	0.44	0.57	0.62	0.57	0.44	0.57	0.62
k_{v_i}	6	3	3	3	4	3	3	3	4
$C_{v_i} \times k_{v_i}$	4.80	1.71	1.32	1.71	2.48	1.71	1.32	1.71	2.48
P_{v_i}	5.60	3.02	2.13	3.02	3.63	3.02	2.13	3.02	3.63
A_{v_i}	5.60	3.02	2.13	3.02	3.63	3.02	2.13	3.02	3.63
L_{v_i}	3	2	1	2	2	2	1	2	2
R_{v_i}	2	2	2	2	2	2	2	2	2

Given a sample network [7] as shown in Figure 3, there are nine nodes. These nodes can be classified into four types: 1) node 1 is the topology center of the network, 2) nodes 5 and 9 are the center of local groups, 3) nodes 2 and 4 are closed to nodes 1 and 5, and nodes 6 and 8 are closed to nodes 1 and 9, 4) nodes 3 and 7 are closed to nodes 5 and 9, respectively. Their closeness centrality, degree and four abilities are shown in Table 2. We can find that nodes have large closeness centrality, if they are in the center of network (e.g., node 1). However, the difference of their closeness centrality is not obvious. The personal importance of node v_i ($C_{v_i} \times k_{v_i}$) can correct the closeness centrality with the degree of nodes. It can be seen that personal importance can better reflect the difference of nodes than closeness centrality. Further, the influence of nodes is reflected obviously by propagation ability. The values of attraction ability are equal to propagation ability. The launch ability ($\beta = 0.8$) and acceptance ability ($\gamma = 2$) are also calculated.

Here, we give an example to explain the steps of the new label propagation mechanism. Assuming node 2 is the updating node.

1) NALPA determines the nodes which can launch labels to node 2. Node 2 has attraction ability $A_{v_2} = 3.02$, thus it can attract nodes 1, 3, ...,9 whose distance to node 2 < 3.02 to spread their labels. However, as the propagation ability of node 7 is smaller than the distances between node 2 and it, it cannot propagate their labels to node 2. Thus, nodes 1, 3, 4, 5, 6, 8, 9 are found to spread labels to node 2.
2) NALPA determines the labels of nodes which are launched and their quantity. Node 1 can spread three labels in the descending order of belonging coefficients,

node 3 can spread one label, and nodes 4, 5, 6, 8 and 9 propagate two labels to node 2.
3) NALPA determines the labels which are accepted by node 2 and their quantity, then node 2 can accept two labels at most.
4) Starting propagation.

E. LABEL INFLUENCE

Existing label propagation methods only consider the weight of labels in a node, we consider that the weight of labels should change when they arrive at other nodes. Following the above example, node 2 can accept two labels, however, it may receive more than two labels. So which labels should be accepted? Thus, label influence is defined to reflect the weight of labels when they arrive at a node, which is different from belonging coefficient $c_j^{v_i}$ reflecting the weight of label $l_j^{v_i}$ in the label set of node v_i. Like radar transmission, when node v_j launches label l to node v_i, the label influence of label l is related to the three factors, the propagation ability of node v_j, the distances between between nodes v_i and v_j and the belonging coefficient of label l. The order of label accepting is determined according to label influence, and we define label influence as

$$LP_{l,v_j \to v_i} = \frac{P_{v_j}}{d_{v_i v_j}} \times c_l^{v_j}. \tag{6}$$

IV. THE PROPOSED ALGORITHM

In this section, we propose NALPA to address the instability and enhance the efficiency. The overall process is first described and then its time complexity is analyzed.

A. ALGORITHM DESCRIPTION

As shown in Figure 2, the procedures of NALPA include initialization, node choice, label launch, label acceptation, termination judgment and post-processing.

Step 1 Initialization

In this step, all nodes are assigned with the initial label, and their node abilities are computed.

1) Set $S = V$, $B(v_i) = \{(l_1^{v_i} = i, c_1^{v_i} = 1)\}$ for $v_i \in V$ and $a = 1$, $t = 1$. Here, S denotes the node set, in which nodes have not been updated.
2) Four node abilities $P_{v_i}, A_{v_i}, L_{v_i}$ and R_{v_i} for $v_i \in V$ are computed, then the nodes in S are ordered in the ascending order of propagation ability.

Step 2 Node choice

In this step, node v_i is chosen to update its labels. Then the nodes that can launch labels to node v_i are determined.

1) Node v_i satisfying $P_{v_i} = min(P_{v_j} | v_j \in S)$ is selected.
2) If node v_j satisfies $d_{v_i v_j} \le A_{v_i}$, then $V_{v_i}^A = V_{v_i}^A \bigcup \{v_j\}$. If there does not exist node v_j satisfying $d_{v_i v_j} \le A_{v_i}$, then $V_{v_i}^A = V_{v_i}^A \bigcup N(v_i)$, where $V_{v_i}^A$ denotes the nodes attracted by node v_i.
3) If node $v_j \in V_{v_i}^A$ and $P_{v_j} \ge d_{v_i v_j}$, then $V_{v_i}^S = V_{v_i}^S \bigcup \{v_j\}$. If $\nexists v_j \in V_{v_i}^A$ satisfies $P_{v_j} \ge d_{v_i v_j}$, then $V_{v_i}^S = V_{v_i}^S \bigcup N(v_i)$, where $V_{v_i}^S$ denotes the nodes that can launch their labels to node v_i.

Step 3 Label Launch

In the step of label launch, node v_j in $V_{v_i}^S$ launches their labels to node v_i in the descending order of belonging coefficient.

1) For node v_j in $V_{v_i}^S$, label l^{v_j} satisfying $c^{v_j} = max(c^{v_k} | (l^{v_k}, c^{v_k}) \in B(v_j))$ is selected, then $B(v_j) = B(v_j) - \{(l^{v_j}, c^{v_j})\}$, $B(v_i) = B(v_i) \bigcup \{(l^{v_j}, LP_{l^{v_j}, v_j \to v_i})\}$, and $n_{v_j}^L = n_{v_j}^L + 1$ until $n_{v_j}^L = L_{v_j}$ or $B(v_j) = \emptyset$, where $n_{v_j}^L$ is the number of labels launched by node v_j. $B(v_j)$ is recovered for next label propagation.
2) If the labels with the same id arrive at node v_i, their label influence are summed.

Step 4 Label Acceptation

After the procedure of label launch, many labels may arrive at node v_i. This step is used to accept useful labels and filter the labels with small label influence.

1) We first sort the labels of node v_i by their label influence, then $B(v_i) = \{(l_1^{v_i}, LP_{l_1^{v_i}}), \ldots, (l_j^{v_i}, LP_{l_j^{v_i}}), \ldots, (l_R^{v_i}, LP_{l_R^{v_i}})\}$ after label launch step, where R is the number of labels received from other nodes. Then node v_i accepts R_{v_i} labels in order. Thus, $B(v_i) = \{(l_1^{v_i}, LP_{l_1^{v_i}}), \ldots, (l_j^{v_i}, LP_{l_j^{v_i}}), \ldots, (l_{R_{v_i}}^{v_i}, LP_{l_{R_{v_i}}^{v_i}})\}$.
2) By normalizing the label influence of labels in $B(v_i)$, we can obtain $B(v_i) = \{(l_1^{v_i}, c_1^{v_i}), \ldots, (l_j^{v_i}, c_j^{v_i}), \ldots, (l_{R_{v_i}}^{v_i}, c_{R_{v_i}}^{v_i})\}$.
3) For label $\left(l_j^{v_i}, c_j^{v_i}\right)$ in $B\left(v_i\right)$, if $c_j^{v_i} < 1/R_{v_i}$, then $B\left(v_i\right) = B\left(v_i\right) - \{\left(l_j^{v_i}, c_j^{v_i}\right)\}$. Then we obtain the updated $B(v_i)$ of node v_i by normalizing again. The updating of labels of node v_i is finished in the tth iteration.
4) If $a = N$, step 5 is executed, else the method sets $a = a + 1$, $S = S - \{v_i\}$ and returns to step 2.

Step 5 Termination Judgment

The proposed algorithm calculates the minimal number set m_t of nodes signed by each community identifier [13], [14]. If $m_t = m_{t-1}$, NALPA goes to step 6 for post-processing, else sets $S = V$, $a = 1$, $t = t + 1$ and returns to step 2.

Step 6 Post-processing

When NALPA stops, nodes containing community label l are allocated to community O_l. Overlapping nodes are allocated to multiple communities.

To understand the label propagation of NALPA intuitively, we present the process of NALPA in the sample network in Figure 4. Firstly, as shown in Figure 4(a), the nine nodes are assigned unique labels with belonging coefficients: (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1) and (9, 1). Second, by calculating four abilities and sorting the nodes in the ascending order of propagation ability, then we can get the updating sequence 3→7→2→6→4→8→5→9→1. As shown in Figure 4(b), node 3 is selected to update its label. The attraction ability of node 3 is 2.13, thus, it can attract nodes 1, 2, 4 and 5 to launch their labels, and the propagation

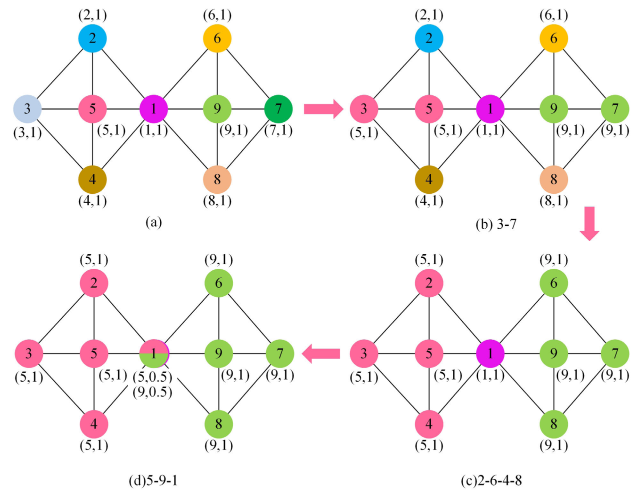


FIGURE 4. Label propagation process of NALPA on the sample network (in the ascending order of propagation ability 3 → 7 → 2 → 6 → 4 → 8 → 5 → 9 → 1).

abilities of nodes 1, 2, 4 and 5 are all larger than the distances between node 3 and them. Thus, nodes 1, 2, 4 and 5 spread (1,1), (2,1), (4,1) and (5,1) to node 3, respectively. When the labels arrive at node 3, we calculate their label influence and obtain node 3: (1, 2.80), (2, 3.02), (4, 3.02) and (5, 3.63). Then node 3 accepts two labels (4, 3.02) and (5, 3.63) (NALAP chooses label 4 randomly from labels 2 and 4 because they have the same label influence). However, the belonging coefficient of label 4 ($3.02/(3.02+3.63) = 0.45$) is smaller than 0.5 ($1/R_{v_i} = 0.5$), then label 4 is filtered. Finally, the label of node 3 is updated as (5, 1). In the same manner, the label of node 7 is updated as (9, 1). As shown in Figure 4(c), node 2 begins to update its label. The attraction ability of node 2 is 3.02, thus, it can attract nodes $1, 3, \ldots, 9$ to launch their labels. However, the propagation ability of node 7 is smaller than the distances between node 2 and it ($P_{v_7} = 2.13 < 3$). Then only nodes 1, 3, 4, 5, 6, 8 and 9 can spread (1, 1), (5, 1), (4, 1), (5, 1), (6, 1), (8, 1) and (9, 1) to node 2, respectively. When the labels arrive at node 2, we compute their label influence and obtain node 2: (1, 5.6), (5, 2.13), (4, 1.51), (5, 3.63), (6, 1.51), (8, 1.51) and (9, 1.815). Then node 3 accepts (1, 5.6) and (5, 5.76). Because the belonging coefficient of label 1 (5.6 / (5.6 + 5.76) = 0.49) is smaller than 0.5 ($1/R_{v_i}$), then label 1 is filtered. Finally, the label of node 2 is updated as (5, 1). In the similar way, nodes 6, 4, 8 update their labels as (9, 1), (5, 1) and (9, 1) in turn, respectively. Finally, as shown in Figure 4(d), the labels of nodes 5 and 9 stay the same. Then the labels of node 1 are updated as (5, 0.5) and (9, 0.5). As a results, nodes 1, 2, 3, 4 and 5 are assigned to a community, and nodes 1, 6, 7, 8 and 9 belong to another community. Here, node 1 belongs to two communities at the same time, in other words, node 1 is an overlapping node which has equal belonging coefficient to both communities. It is worth that NALAP only needs one iteration to obtain community results in the sample network. As a contrast, LPANNI [7] needs three iterations in the same network, which illustrates that NALPA can obtain satisfying results and converge fast for using more node and label information.

B. COMPLEXITY ANALYSIS

The time complexity of NALPA is estimated as follows.

1) Initialization: Initializing nodes with unique labels and computing node abilities needs time $O(N)$. Quick sort algorithm is used for sorting nodes by propagation ability with time $O(NlogN)$. As a result, initialization takes time $O(NlogN)$.
2) Node choice: Choosing a node to update its label and determining nodes to launch labels take constant time.
3) Label launch: Determining node set $V_{v_i}^S$ costs time $O(n_1)+O(|V_{v_i}^A|)$ and launching labels to node v_i takes the worst time $O(n_2|V_{v_i}^S|)$, where n_1 is the maximum number of nodes attracted by node v_i, and n_2 is the maximum number of labels of nodes in $V_{v_i}^S$. In general, $n_1 \ll N$ and $n_2 \ll N$. Thus, label launch takes constant time.
4) Label acceptation: Accepting labels costs time $O(n_3)$, where n_3 is the number of labels arrived at node v_i. In general, $n_3 \ll N$. Thus, label acceptation uses constant time.
5) Termination judgment and post-processing: As same as COPRA, the former takes time $O(\gamma N)$, and the latter takes time $O((\gamma^3+1)N + \gamma(N+M))$ [13].

For the updating of the labels of node v_i, steps 2-4 need constant time. Thus, updating the labels of all nodes in one iteration needs time $O(N)$. Then the time complexity of NALPA is $O(NlogN + \gamma M + (\gamma^3 + 2\gamma + 1 + t)N)$. Thus, the time complexity of NALPA is near $O(NlogN + c_1N + c_2M)$, where c_1 and c_2 are positive integer.

V. EXPERIMENTAL SETUPS

In this paper, extensive experiments are conducted on Intel Core i3-4370 CPU running at 3.80 GHz with 8 GB memory to evaluate the effectiveness of NALPA. Each algorithm independently runs 50 times. The synthetic and real-world networks are introduced firstly. Then the baselines and evaluation criteria are described. Finally, we conduct the parameter analysis.

A. DATASETS

In this paper, 42 synthetic and 14 real-world networks are adopted to evaluate the performance of the proposed algorithm. Synthetic networks are generated based on LFR benchmark which imports heterogeneity into degree and community size distributions governed by power laws with exponents τ_1 and τ_2, respectively, [27]. In experiments, the synthetic networks are generated with some parameters described in Table 3. Mixing parameter μ denotes the expected fraction of edges of a node connecting to other communities. To generate the networks with overlapping communities, we set the number of overlapping nodes $O_n \geq 1$ and assign nodes to $O_m \geq 1$ communities.

The synthetic networks are given in Table 4. We use the networks with $\tau_1 = 2$ and $\tau_2 = 1$ [27]. LFR-1-16 are

TABLE 3. Parameters for synthetic networks.

Parameter	Meaning
N	node number
$<k>$	average degree
$maxk$	the maximum degree of nodes
$minc$	the minimum of community size
$maxc$	the maximum of community size
μ	mixing parameter
O_n	the number of overlapping nodes
O_m	the number of communities that each overlapping node belongs to

TABLE 4. Description of synthetic networks.

Network	N	$<k>$	$maxk$	$minc$	$maxc$	μ	O_n	O_m
LFR-1	5000	10	50	20	100	0.1	0	1
LFR-2	5000	10	50	20	100	0.3	0	1
LFR-3-9	5000	10	50	20	100	0.1	500	2-8
LFR-10-16	5000	10	50	20	100	0.3	500	2-8
LFR-17	10000	10	50	20	100	0.1	0	1
LFR-18	10000	10	50	20	100	0.3	0	1
LFR-19-25	10000	10	50	20	100	0.1	500	2-8
LFR-26-32	10000	10	50	20	100	0.3	1000	2-8
LFR-33-37	10000	10	50	20	100	0.1	400-2000	2
LFR-38-42	10000	5-30	50	20	100	0.1	1000	2

TABLE 5. Description of real-world networks.

Network	Reference	N	M	c	$<k>$	dia	cc
Karate	[28]	34	78	2	4.588	5	0.588
Dolphins	[29]	62	159	2	5.129	8	0.303
Polbook	[30]	105	441	2	8.400	7	0.488
Football	[31]	115	615	12	10.661	4	0.403
Lesmis	[32]	77	254	-	6.579	5	0.736
Jazz	[33]	198	2742	-	27.697	6	0.633
Email	[34]	1133	5451	-	9.624	8	0.255
Netscience	[35]	1589	2742	-	3.451	17	0.878
Power	[36]	4941	6594	-	2.669	46	0.107
PGP	[37]	10680	24316	-	4.554	24	0.440
Cond2003	[38]	31163	120029	-	7.703	16	0.723
Enron	[39]	36692	183831	-	10.020	13	0.716
Cond2005	[38]	40421	175693	-	8.693	18	0.719
Amazon	[40]	334863	925872	-	5.530	44	0.396

"c" denotes the number of communities, "dia" denotes the diameter of networks, "cc" denotes the average clustering coefficient of networks, Amazon is acquired from http://snap.stanford.edu/data/com-Amazon.html.

the networks with $N = 5000$, in which LFR-1-2 are the networks with non-overlapping communities, LFR-3-16 are the networks with overlapping communities where O_n is 10% of N and O_m varies from 2 to 8 indicating the diversity of overlapping nodes. LFR-17-32 are the networks with $N = 10000$, whose other parameters are the same as the corresponding networks with $N = 5000$. For LFR-33-37, O_n varies from 400 to 2000 with an interval 400. For LFR-38-42, $<k>$ varies from 5 to 30 with an interval 5. We select the real-world networks with known or unknown true communities, then we sort them according to the number of nodes, which are listed in Table 5.

B. BASELINE ALGORITHMS

To give a well-rounded performance comparison with state-of-the-art algorithms, we compare our method with nine community detection algorithms based on label propagation listed below.

- LPA [12] is a fast algorithm for discovering non-overlapping communities by updating the label of each node to the most popular among its neighbors.
- COPRA [13] is an efficient model for discovering overlapping communities in networks by assigning multiple labels with belonging coefficients to a node.
- SLPA [25] is an overlapping community detection approach by allocating nodes memory spaces to store received labels.
- DLPA$^+$ [15] computes the confidence of neighbors and the weight of labels to discover overlapping communities.
- WLPA [14] is a label weight based method to discover overlapping community, which spreads the labels with large weight in label propagation process.
- LINSIA [17] is designed for detecting overlapping communities based on node importance and label influence.
- LPA_NI [18] is also proposed for discovering overlapping communities based on node importance and label influence.
- NGLPA [19] is proposed for discovering communities based on node gravitation and node similarity.
- LPANNI [7] is designed for discovering overlapping communities based on node importance, node similarity and neighbor node influence.

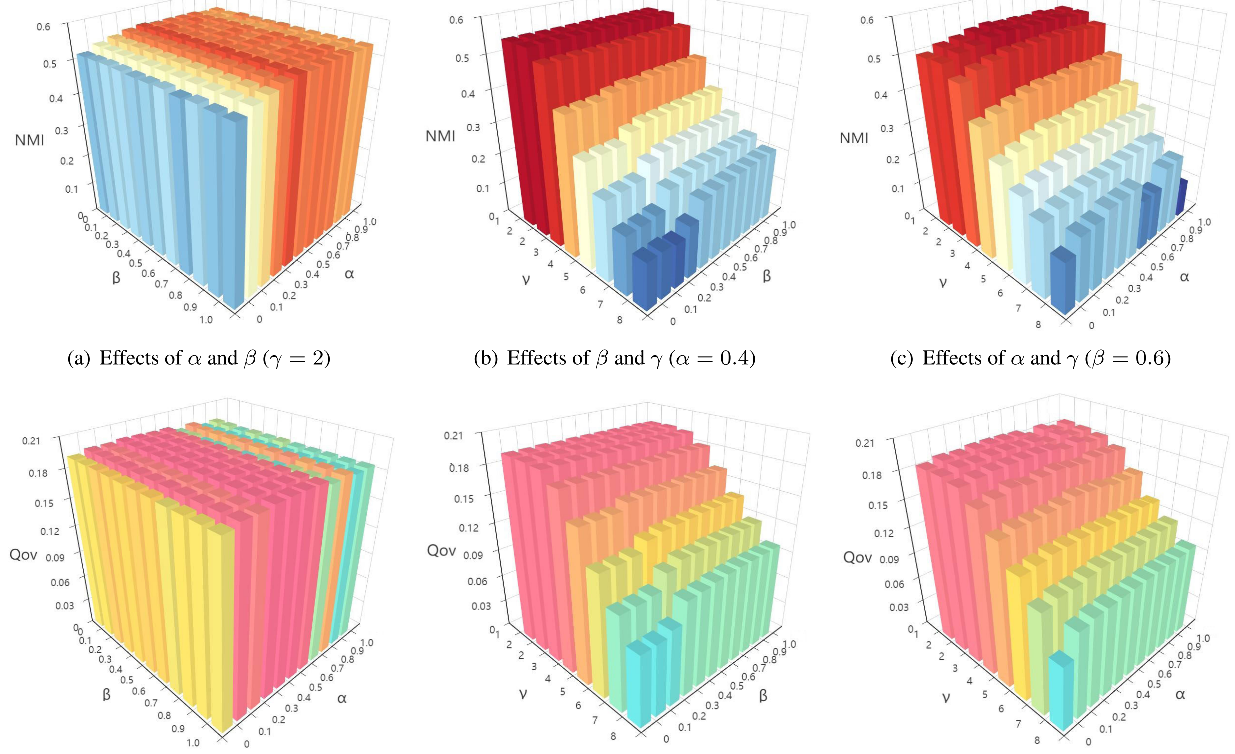
TABLE 6. Time complexity of methods.

Algorithm	Complexity
LPA	$O(tM+N)$
COPRA	$O(t\gamma Mlog(\gamma M/N)+\gamma M+(\gamma^3+\gamma)N)$
SLPA	$O(tM+N)$
DLPA$^+$	$O(NlogN+tM+N)$
WLPA	$O(t\gamma Mlog(\gamma M/N)+\gamma M+(\gamma^3+\gamma)N)$
LINSIA	$O(NlogN+tM+N)$
LPA_NI	$O(NlogN+2tM+3N)$
NGLPA	$O((N+1)logN+tM+(2+maxk)N)$
LPANNI	$O(NlogN+(<k>^{\alpha-1}+1)M+(<k>^2+(t+1)<k>+1)N)$
NALPA	$O(NlogN+\gamma M+(\gamma^3+2\gamma+1+t)N)$

Here, we present the time complexity of our method and baselines, as shown in Table 6. We can find that their time complexity can be classified as two types, 1) the time complexity is $O(h_1M + h_2N)$ if the methods do not need to sort nodes, 2) the time complexity is $O(NlogN + h_3M + h_4N)$ if the methods sore nodes in prescribed rules, where $h_1, \ldots, h_4$ are different coefficients for different methods.

C. EVALUATION CRITERIA

We use five criteria, overlapping modularity, normalized mutual information, precision, recall and F-score, to evaluate the quality of detected communities. Normalized Mutual Information (NMI) [41] is used to show the difference between the results of experimental algorithms and true communities. The larger the value of NMI, the smaller the



(a) Effects of α and β ($\gamma = 2$) (b) Effects of β and γ ($\alpha = 0.4$) (c) Effects of α and γ ($\beta = 0.6$)

(d) Effects of α and β ($\gamma = 2$) (e) Effects of β and γ ($\alpha = 0.4$) (f) Effects of α and γ ($\beta = 0.6$)

FIGURE 5. Results of parameter analysis.

difference. Overlapping modularity (Q_{OV}), the extension of Newman's modularity [42], reflects the quality of divisions assessed by the relative density of edges within communities and between communities [43]. As the Q_{OV} function, we adopt $f(x) = 60x-30$ [13]. Like NIBLPA, we use *F-score* (F) [16], [44], also called *F-measure* to quantify the accuracy of community detection, which is defined as

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{7}$$

where *Precision* (P) and *Recall* (R) are represented as

$$Precision = \frac{|S \cap T|}{|S|}, \tag{8}$$

$$Recall = \frac{|S \cap T|}{|T|}. \tag{9}$$

T is the set of node pairs (v_i, v_j) where nodes v_i and v_j belong to the same communities in the ground truth, and S is the set of node pairs that belong to the same communities generated by community detection algorithms. Then $S \cap T$ represents the intersection of node pairs of the ground truth and the community results. *F-score* is the comprehensive measurement of *Precision* and *Recall*. The larger the value of *F-score*, the more accurate the community results.

If the true communities of networks are known, five criteria are both adopted, otherwise only Q_{OV} is adopted. In addition, we adopt Running Time (RT) to evaluate the efficiency of methods. To intuitively compare the comprehensive performance of methods, we compute the rank of different algorithms in each network and calculate their average rank. The smaller the average rank, the better the performance of methods.

TABLE 7. LFR-0 for parameter analysis.

Network	N	<k>	*maxk*	*minc*	*maxc*	μ	O_n	O_m
LFR-0	5000	10	50	20	100	0.5	500	6

D. PARAMETER SETTINGS

Here, we explore the impact of parameters α, β and γ on NALPA, in which α controls the effect of neighbors' importance for propagation ability, β controls the effect of propagation ability for launch ability and γ is the accepting threshold. LFR-0 with large μ and O_m is used for parameter analysis, as shown in Table 7, so the algorithms cannot discover communities easily because the communities of networks with large μ and O_m become intricate. If NALPA with some parameters can obtain satisfying communities in LFR-0, we deduce that NALPA can obtain desirable partition for the networks with small μ and O_m. We analyze the effects of two parameters by fixing another parameter to investigate how they affect community results. Parameters α and β vary from 0 to 1 with interval 0.1, and γ varies from 1 to 8 with interval 1. The effects of α, β and γ in terms of *NMI* and Q_{OV} are presented in Figure 5.

As shown in Figure 5(a), when $\alpha \in [0, 0.4]$, *NMI* rises with the increase of α. When $\alpha \in [0.4, 1]$, the results of *NMI* keep steady. We can find that NALPA with $\alpha = 0.4$ outputs the best *NMI*, which is beneficial for improving the performance. Thus, major personal importance and some portion of neighbor importance can better reflect the influence of nodes and are helpful for community detection. As shown in Figure 5(b), when $\beta \in [0, 0.6]$, *NMI* rises with the increase of β, and *NMI* keeps steady when $\beta \in [0.6, 1]$, which means that launching labels with the number equaling to a large proportion of propagation ability can obtain good results. Thus, we choose $\beta = 0.6$. It can be seen that parameter γ has large influence on the results of *NMI*, as shown in Figures 5(b) and 5(c), which decline with the increase of γ. When nodes belong to more than three communities, the results become poor, since nodes are not closed in each community for there nodes possibly and the methods cannot detect accurate overlapping nodes in multiple communities easily. By observing the trends, we can conclude that NALPA obtains the best *NMI* with $\gamma = 1$ and 2, then we choose $\gamma = 2$ for overlapping community detection. By discussing these parameters, we infer that NALPA achieves the best *NMI* when $\alpha = 0.4$, $\beta = 0.6$ and $\gamma = 2$ for LFR-0. In terms of Q_{OV}, NALPA also obtains the best result with the same parameter values for LFR-0, as shown in Figures 5(d)-5(f). Therefore, we choose NALPA with $\alpha = 0.4$, $\beta = 0.6$ and $\gamma = 2$ to compare with other algorithms.

In order to obtain good community results, the parameters of the baseline algorithms with tunable parameters, including COPRA, SLPA and DLPA$^+$, WLPA and LPANNI, need to be adjusted for different networks. So we adjust and choose the parameters in following experiments to gain good results of these algorithms. Specifically, the baseline algorithms with tunable parameters using the following parameter settings:

1) For COPRA and WLPA, maximum label number γ of each node varies from 1 to 8 [13], [14].
2) For SLPA, probability threshold r varies from 0.01 to 1 with interval 0.01 and iteration time t sets to 100 [25].
3) For DLPA$^+$, inflation factor varies from 1 to 8 and iteration time t also sets to 100 [15].
4) For LPANNI, path length threshold α sets to 3 [7].

The determined parameters are as shown in Table 8.

VI. EXPERIMENTAL RESULTS

A. RESULTS FOR SYNTHETIC NETWORKS

The experimental results for the synthetic networks with non-overlapping communities are shown in Table 9. The number in bracket is the rank of methods for each network, and the average rank is shown at the bottom. The results for the synthetic networks with overlapping communities are shown in Figure 6 ($N = 5000$, $\mu = 0.1$ or 0.3, O_m changes), Figure 7 ($N = 10000$, $\mu = 0.1$ or 0.3, O_m changes), Figure 8 ($N = 10000$, O_n changes) and Figure 9 ($N = 10000$, $<k>$ changes). The average rank of methods for LFR-3-16 and LFR-19-32 is shown in Table 10. The curves of experimental results of partial methods are overlapped to a certain extent.

TABLE 8. Parameter values of baseline algorithms.

Network	COPRA	SLPA	DLPA$^+$	WLPA	LPANNI
LFR-1-42	2	0.45,100	3,100	2	3
Karate	3	0.33,100	3,100	3	3
Dolphin	4	0.45,100	3,100	4	3
Polbook	2	0.45,100	3,100	2	3
Football	2	0.45,100	3,100	2	3
Lesmis	2	0.45,100	3,100	2	3
Jazz	1	0.45,100	3,100	1	3
Email	2	0.45,100	3,100	2	3
Netscience	6	0.45,100	3,100	6	3
Power	2	0.45,100	3,100	2	3
PGP	11	0.45,100	3,100	11	3
Cond2003	2	0.45,100	3,100	2	3
Enron	2	0.45,100	3,100	2	3
Cond2005	2	0.45,100	3,100	2	3
Amazon	2	0.45,100	3,100	2	3

As shown in Table 9, we can find that majority results for LFR-1 are better than the results for LFR-2 obtained by each algorithm, and this phenomenon is similar for LFR-17 and LFR-18 because communities become more intricate when μ varies from 0.1 to 0.3. Community detection algorithms discover communities more hardly, then corresponding evaluation indexes decline. In terms of *NMI*, SLPA and WLPA obtain the best result for LFR-1 and LFR-2, respectively. NALPA ranks the first for LFR-17 and LFR-18. LINSIA obtains the worst results in these networks. Although NALPA ranks the third only to SLPA and LPA_NI for LFR-1 and the second only to WLPA for LFR-2, the difference with the optimal value is small. In terms of Q_{OV}, NGLPA and NALPA outperforms other methods in LFR-1-2 and LFR-17-18, respectively. According to the average rank, NGLPA ranks the first and our method ranks the second, on the contrary, LINSIA also performs the worst. As for *Precision*, WLPA obtains the best result for LFR-1 and NALPA ranks the first for other networks. LINSIA also obtains the worst. But for *Recall*, LINSIA ranks the second. Meanwhile, NGLPA gets the best *Recall*. It can be seen that, some algorithms have imbalanced *Precision* and *Recall*. There are usually two reasons: over-detection and under-detection. For example, LINSIA has low *Precision* but high *Recall*, which indicates that it has the problem of over-detection. Although the *Recall* of LINSIA is high, however, its *Precision* is relatively low in these networks, then its *F-score* is low. WLPA has high *Precision* but low *Recall*, which may be caused by under-detection. In terms of *F-score*, the *Precision* and *Recall* of NALPA are high and balanced, then its *F-score* is the best. As for *RT*, LPA needs the least running time and the efficiency of NALPA is near LPA. In the updating process of NALPA, nodes can attract more nodes not just neighbors to spread their multiple labels, and labels are filtered by label influence, then NALPA can utilize more network topology and label information to reduce iteration times. Among baselines, WLPA is tied for second with NALPA, on the contrary, the efficiency of SLPA is the worst because it requires 100 iterations. As for stability, the standard deviation of LINSIA is 0, in other

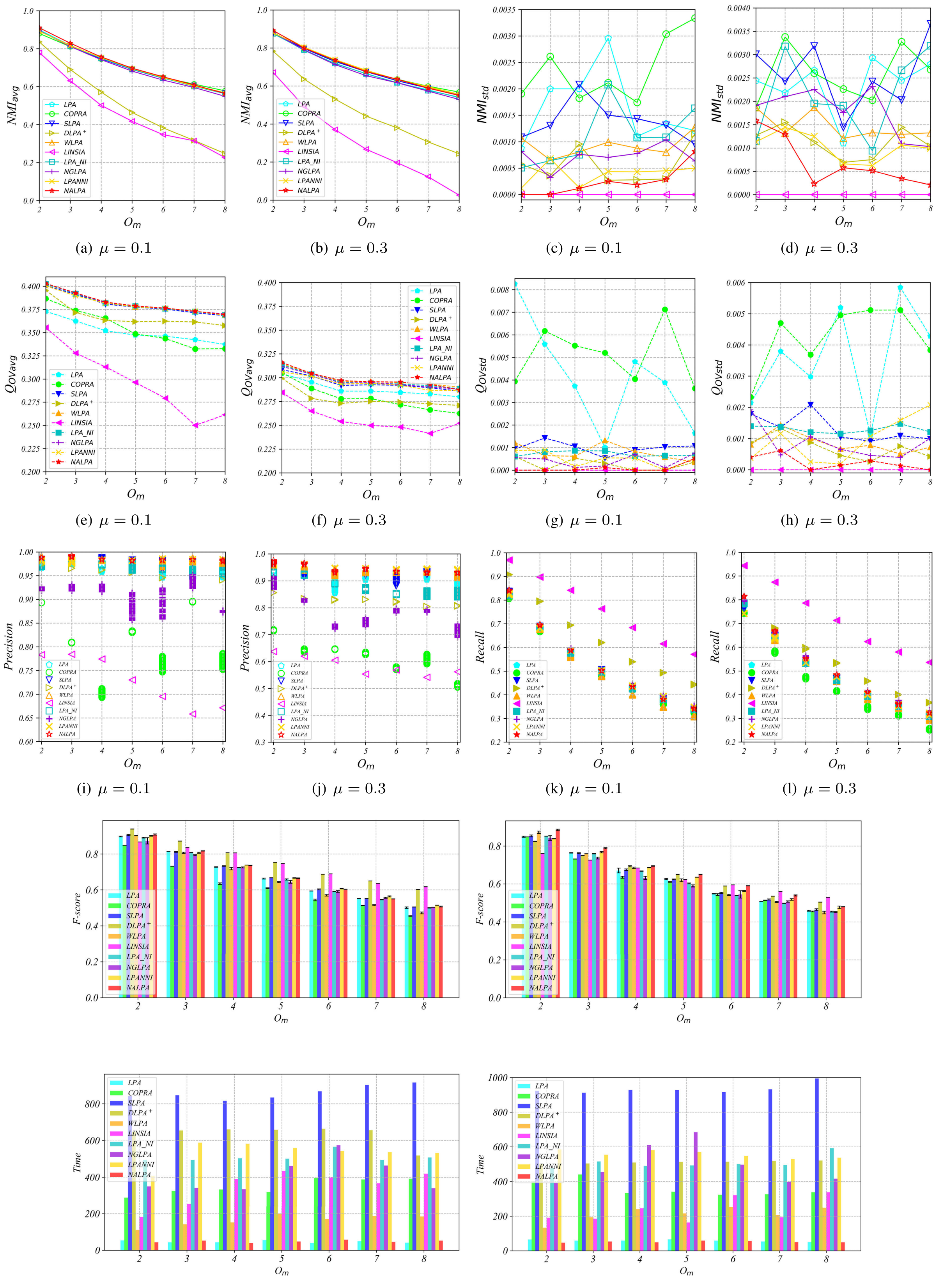
TABLE 9. Results for synthetic networks with non-overlapping communities.

Criterion	Network	LPA	COPRA	SLPA	DLPA+	WLPA	LINSIA	LPA_NI	NGLPA	LPANNI	NALPA
NMI_{avg}	LFR-1	0.9730 (9)	0.9853 (8)	**0.9994 (1)**	0.9887 (7)	0.9980 (3)	0.8813 (10)	0.9987 (2)	0.9973 (6)	0.9980 (3)	0.9980 (3)
	LFR-2	0.9939 (4)	0.9859 (7)	0.9931 (5)	0.9414 (9)	**0.9979 (1)**	0.8267 (10)	0.9847 (8)	0.9916 (6)	0.9951 (2)	0.9951 (2)
	LFR-17	0.9990 (4)	0.9949 (7)	0.9995 (2)	0.9937 (9)	0.9939 (8)	0.9008 (10)	0.9981 (6)	0.9989 (5)	0.9994 (3)	**0.9999 (1)**
	LFR-18	0.9972 (2)	0.9569 (8)	0.9940 (4)	0.9439 (9)	0.9858 (7)	0.8527 (10)	0.9872 (6)	0.9950 (3)	0.9934 (5)	**0.9980 (1)**
	Average Rank	4.75	7.50	3.00	8.50	4.75	10.00	5.50	5.00	3.25	**1.75**
NMI_{std}	LFR-1	0.0050 (10)	0.0046 (9)	0.0005 (3)	0.0013 (7)	0.0011 (6)	**0.0000 (1)**	0.0008 (4)	0.0013 (7)	0.0009 (5)	**0.0000 (1)**
	LFR-2	0.0015 (6)	0.0041 (9)	0.0035 (8)	0.0015 (6)	0.0011 (3)	**0.0000 (1)**	0.0012 (4)	0.0047 (10)	0.0001 (2)	0.0013 (5)
	LFR-17	0.0003 (5)	0.0006 (9)	0.0003 (5)	0.0005 (8)	0.0020 (10)	**0.0000 (1)**	0.0001 (3)	0.0003 (5)	0.0001 (3)	**0.0000 (1)**
	LFR-18	0.0004 (5)	0.0001 (2)	0.0014 (10)	0.0003 (4)	0.0001 (2)	**0.0000 (1)**	0.0004 (5)	0.0009 (9)	0.0008 (8)	0.0004 (5)
	Average Rank	6.50	7.25	6.50	6.25	5.25	**1.00**	4.00	7.75	4.50	3.00
Q_{OVavg}	LFR-1	0.4035 (9)	0.4259 (8)	0.4467 (3)	0.4423 (7)	0.4443 (6)	0.3221 (10)	0.4467 (4)	**0.4473 (1)**	0.4463 (5)	0.4472 (2)
	LFR-2	0.3430 (6)	0.3353 (9)	0.3437 (4)	0.3381 (7)	0.3366 (8)	0.3107 (10)	0.3437 (4)	**0.3468 (1)**	0.3462 (2)	0.3438 (3)
	LFR-17	0.4469 (7)	0.4371 (9)	0.4479 (5)	0.4469 (7)	0.4480 (4)	0.2911 (10)	0.4482 (3)	0.4485 (2)	0.4477 (6)	**0.4488 (1)**
	LFR-18	0.3476 (3)	0.3304 (9)	0.3453 (6)	0.3392 (8)	0.3464 (5)	0.3216 (10)	0.3471 (4)	0.3482 (2)	0.3446 (7)	**0.3488 (1)**
	Average Rank	6.25	8.75	4.50	7.25	5.75	10.00	3.75	**1.50**	5.00	1.75
Q_{OVstd}	LFR-1	0.0022 (9)	0.0078 (10)	0.0012 (5)	0.0016 (7)	0.0017 (8)	**0.0000 (1)**	0.0002 (3)	0.0002 (3)	0.0012 (5)	**0.0000 (1)**
	LFR-2	0.0002 (4)	0.0028 (10)	0.0019 (9)	0.0007 (5)	0.0014 (8)	**0.0000 (1)**	0.0011 (7)	0.0008 (6)	0.0001 (2)	0.0001 (2)
	LFR-17	0.0006 (9)	0.0034 (10)	0.0005 (8)	0.0000 (1)	0.0004 (7)	**0.0000 (1)**	0.0002 (4)	0.0002 (4)	0.0003 (6)	**0.0000 (1)**
	LFR-18	0.0005 (7)	0.0005 (7)	0.0011 (10)	0.0003 (5)	0.0004 (6)	**0.0000 (1)**	0.0009 (9)	0.0001 (2)	0.0002 (4)	0.0001 (2)
	Average Rank	7.25	9.25	8.00	4.50	7.25	**1.00**	5.75	3.75	4.25	1.50
P_{avg}	LFR-1	0.9737 (8)	0.9948 (6)	0.9978 (3)	0.9886 (7)	**0.9999 (1)**	0.6205 (10)	0.9969 (5)	0.9690 (9)	0.9998 (2)	0.9978 (3)
	LFR-2	0.9797 (6)	0.9926 (5)	0.9943 (4)	0.9210 (9)	0.9966 (2)	0.6719 (10)	0.9739 (7)	0.9288 (8)	0.9950 (3)	**0.9971 (1)**
	LFR-17	1.0000 (1)	0.9926 (7)	0.9999 (3)	0.9907 (8)	0.9999 (3)	0.5715 (10)	0.9958 (6)	0.9895 (9)	0.9999 (3)	**1.0000 (1)**
	LFR-18	0.9939 (3)	0.8841 (9)	0.9929 (5)	0.9074 (8)	0.9984 (2)	0.6945 (10)	0.9625 (6)	0.9459 (7)	0.9934 (4)	**0.9993 (1)**
	Average Rank	4.50	6.75	3.75	8.00	2.00	10.00	6.00	8.25	3.00	**1.50**
P_{std}	LFR-1	0.0050 (9)	0.0001 (3)	0.0021 (8)	0.0001 (3)	0.0000 (1)	**0.0000 (1)**	0.0005 (7)	0.0176 (10)	0.0001 (3)	0.0001 (3)
	LFR-2	0.0151 (9)	0.0008 (6)	0.0015 (7)	0.0001 (2)	0.0002 (5)	**0.0000 (1)**	0.0058 (8)	0.0390 (10)	0.0001 (2)	0.0001 (2)
	LFR-17	**0.0000 (1)**	0.0008 (9)	**0.0000 (1)**	**0.0000 (1)**	**0.0000 (1)**	**0.0000 (1)**	0.0002 (8)	0.0041 (10)	**0.0000 (1)**	**0.0000 (1)**
	LFR-18	0.0034 (10)	0.0007 (6)	0.0021 (8)	0.0006 (5)	0.0003 (4)	**0.0000 (1)**	0.0001 (2)	0.0031 (9)	0.0008 (7)	0.0002 (3)
	Average Rank	7.25	6.00	6.00	2.75	2.75	**1.00**	6.25	9.75	3.25	2.25
R_{avg}	LFR-1	0.9604 (9)	0.9703 (8)	0.9991 (2)	0.9754 (7)	0.9574 (10)	0.9938 (6)	0.9968 (4)	**0.9996 (1)**	0.9954 (5)	0.9991 (2)
	LFR-2	0.9789 (6)	0.9885 (4)	0.9627 (9)	0.9277 (10)	0.9789 (6)	**0.9978 (1)**	0.9771 (8)	0.9919 (2)	0.9852 (5)	0.9907 (3)
	LFR-17	0.9925 (8)	0.9885 (9)	0.9985 (3)	0.9931 (7)	0.9834 (10)	0.9993 (2)	0.9964 (6)	**0.9996 (1)**	0.9965 (5)	0.9971 (4)
	LFR-18	0.9892 (3)	0.9175 (10)	0.9768 (5)	0.9350 (9)	0.9593 (8)	**0.9964 (1)**	0.9749 (6)	0.9958 (2)	0.9729 (7)	0.9875 (4)
	Average Rank	6.50	7.75	4.75	8.25	8.50	2.50	6.00	**1.50**	5.50	3.25
R_{std}	LFR-1	0.0050 (9)	0.0035 (7)	0.0003 (4)	**0.0000 (1)**	0.0106 (10)	**0.0000 (1)**	0.0007 (6)	0.0003 (4)	0.0045 (8)	0.0001 (3)
	LFR-2	0.0014 (6)	0.0046 (7)	0.0199 (10)	0.0002 (3)	0.0063 (8)	**0.0000 (1)**	0.0006 (5)	0.0064 (9)	0.0004 (4)	0.0001 (2)
	LFR-17	0.0028 (8)	0.0046 (9)	0.0011 (6)	**0.0000 (1)**	0.0059 (10)	**0.0000 (1)**	0.0003 (5)	**0.0000 (1)**	0.0016 (7)	**0.0000 (1)**
	LFR-18	0.0042 (7)	0.0022 (6)	0.0067 (10)	0.0009 (3)	0.0043 (8)	**0.0000 (1)**	0.0018 (5)	0.0017 (4)	0.0057 (9)	0.0003 (2)
	Average Rank	7.50	7.25	7.50	2.00	9.00	**1.00**	5.25	4.50	7.00	2.00
F_{avg}	LFR-1	0.9670 (9)	0.9824 (6)	0.9985 (1)	0.9820 (7)	0.9782 (8)	0.7640 (10)	0.9968 (4)	0.9840 (5)	0.9976 (3)	**0.9985 (1)**
	LFR-2	0.9793 (5)	0.9905 (2)	0.9782 (6)	0.9243 (9)	0.9877 (4)	0.8030 (10)	0.9755 (7)	0.9590 (8)	0.9901 (3)	**0.9939 (1)**
	LFR-17	0.9962 (4)	0.9905 (9)	**0.9992 (1)**	0.9919 (7)	0.9916 (8)	0.7272 (10)	0.9961 (5)	0.9945 (6)	0.9982 (3)	0.9985 (2)
	LFR-18	0.9915 (2)	0.9005 (9)	0.9848 (3)	0.9210 (8)	0.9784 (5)	0.8185 (10)	0.9686 (7)	0.9702 (6)	0.9830 (4)	**0.9934 (1)**
	Average Rank	5.00	6.50	2.75	7.75	6.25	10.00	5.75	6.25	3.25	**1.25**
F_{std}	LFR-1	0.0050 (8)	0.0019 (6)	0.0010 (5)	0.0006 (3)	0.0055 (9)	**0.0000 (1)**	0.0006 (3)	0.0092 (10)	0.0023 (7)	0.0001 (2)
	LFR-2	0.0083 (8)	0.0023 (5)	0.0110 (9)	0.0004 (4)	0.0031 (7)	**0.0000 (1)**	0.0025 (6)	0.0238 (10)	0.0002 (3)	0.0001 (2)
	LFR-17	0.0014 (7)	0.0023 (9)	0.0006 (5)	**0.0000 (1)**	0.0030 (10)	**0.0000 (1)**	0.0003 (4)	0.0020 (8)	0.0008 (6)	**0.0000 (1)**
	LFR-18	0.0004 (3)	0.0007 (4)	0.0045 (10)	0.0007 (4)	0.0024 (8)	**0.0000 (1)**	0.0008 (6)	0.0008 (6)	0.0033 (9)	0.0001 (2)
	Average Rank	6.50	6.00	7.25	3.00	8.50	**1.00**	4.75	8.50	6.25	1.75
Time(s)	LFR-1	42.2405 (2)	130.1102 (4)	847.8163 (10)	462.4332 (8)	48.1446 (3)	175.0851 (5)	494.3510 (9)	346.1910 (6)	449.2735 (7)	**28.7742 (1)**
	LFR-2	**48.5255 (1)**	189.9801 (6)	941.3162 (10)	77.2122 (4)	65.5102 (3)	184.1243 (5)	480.5123 (8)	383.0065 (7)	561.6215 (9)	52.0517 (2)
	LFR-17	**90.9030 (1)**	922.0710 (7)	1761.5080 (10)	113.0830 (3)	102.6380 (2)	283.9185 (5)	1118.1175 (9)	763.9650 (6)	944.3705 (8)	155.3865 (4)
	LFR-18	**115.9075 (1)**	1435.7225 (9)	1469.3830 (10)	1247.7765 (8)	125.5000 (2)	425.0810 (4)	1056.8120 (7)	705.0705 (5)	975.9230 (6)	197.5745 (3)
	Average Rank	**1.25**	6.50	10.00	5.75	2.50	4.75	8.25	6.00	7.50	2.50

words, LINSIA is stable. If the number of the most important labels of a node is more than one, LINSIA spreads all labels to other nodes, which does avoid the instability problem but makes LINSIA discover inaccurate communities and waste efficiency. NALPA ranks the second in terms of standard deviation that means NALPA is more stable than other methods except LINSIA because NALPA determines the node order of label updating according to propagation ability, the order of label launching according to belonging coefficient and the order of label accepting according to label influence. However, when the number of the most important labels of a node is more than its launch ability, NALPA randomly spreads these labels, which also causes the instability problem. Among baselines, the stability of LPA, COPRA, SLPA and NGLPA is worse than some methods.

For the synthetic networks with overlapping communities when $N = 5000$, as shown in Figure 6, the results of *NMI*, Q_{OV}, *Recall* and *F-score* reduce gradually with the increase of O_m, and the values of *NMI* and Q_{OV} when $\mu = 0.3$ are smaller than those when $\mu = 0.1$, which also illustrates that the communities become more intricate with the increase of O_m and μ. On the other hand, the variation of O_m has little influence on running time.

In terms of *NMI*, as shown in Figures 6(a) and 6(b), the algorithms except DLPA$^+$ and LINSIA behave similarly and obtain good results, and NALPA gains the highest average rank as shown in Table 10. Among them, LPA updates the label of a node to the most frequent label of neighbors. COPRA assigns the labels with belonging coefficients to measure the membership of nodes in different communities. SLPA records received labels and considers the communities with the largest frequency label as nodes' communities. WLPA defines the weight of labels to reflect the ownership of nodes. LPA_NI identifies the membership of nodes by node importance and label influence from expert knowledge. NGLPA discovers the relationship among



(a) $\mu = 0.1$ (b) $\mu = 0.3$ (c) $\mu = 0.1$ (d) $\mu = 0.3$

(e) $\mu = 0.1$ (f) $\mu = 0.3$ (g) $\mu = 0.1$ (h) $\mu = 0.3$

(i) $\mu = 0.1$ (j) $\mu = 0.3$ (k) $\mu = 0.1$ (l) $\mu = 0.3$

(o) $\mu = 0.1$ (p) $\mu = 0.3$

FIGURE 6. Results for the synthetic networks with overlapping communities when $N = 5000$.

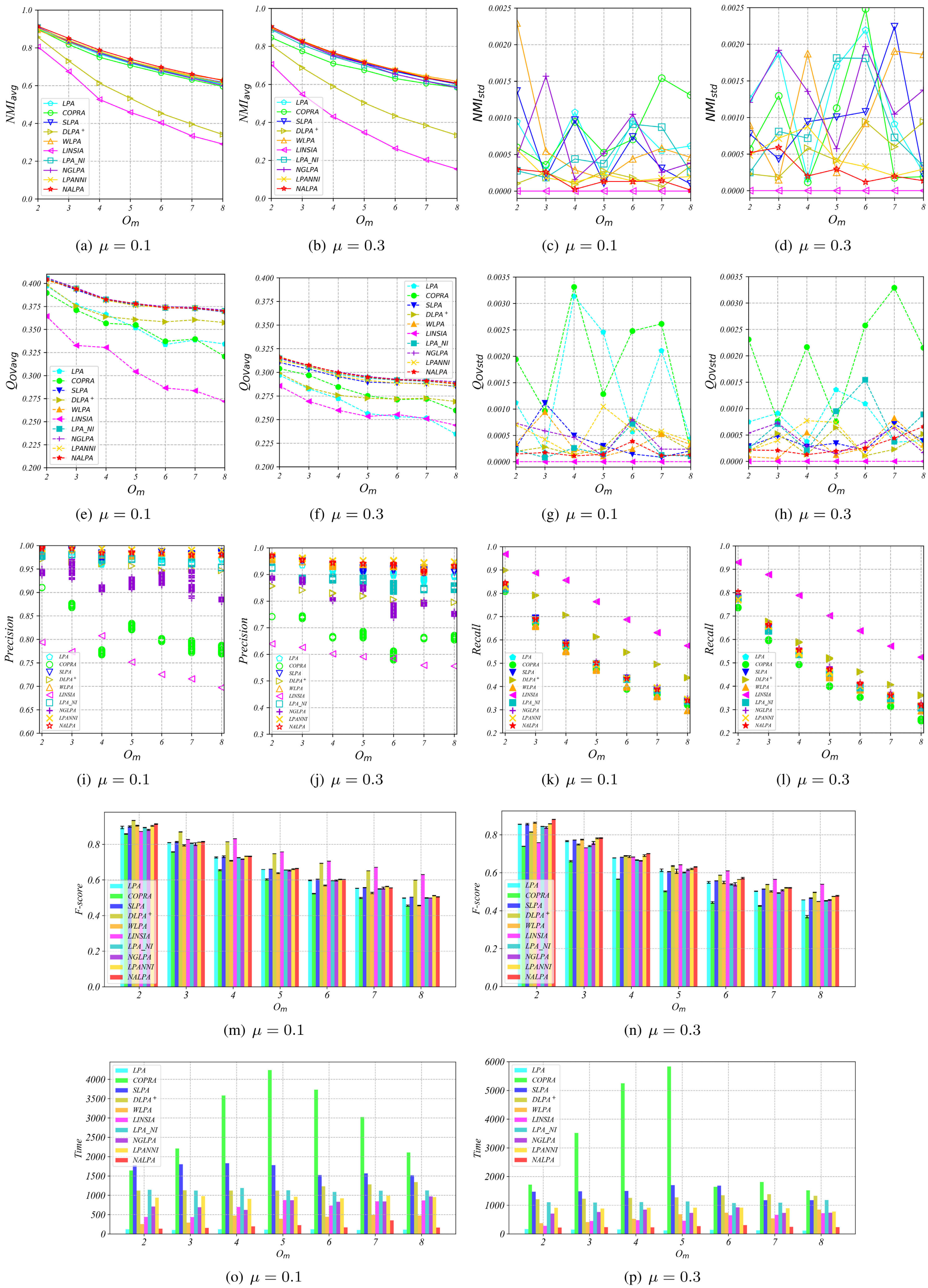


FIGURE 7. **Results for the synthetic networks with overlapping communities when $N = 10000$.**

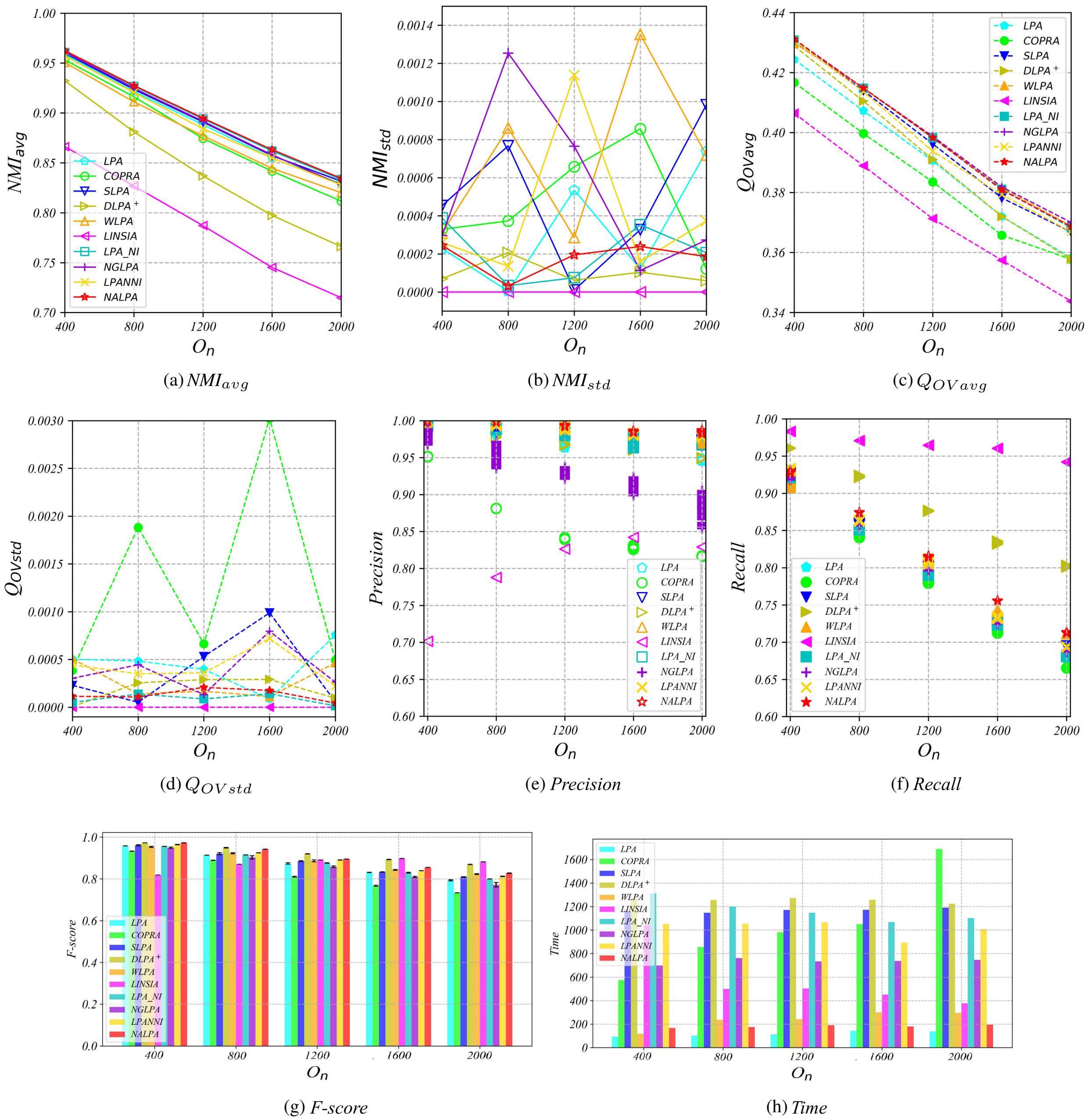
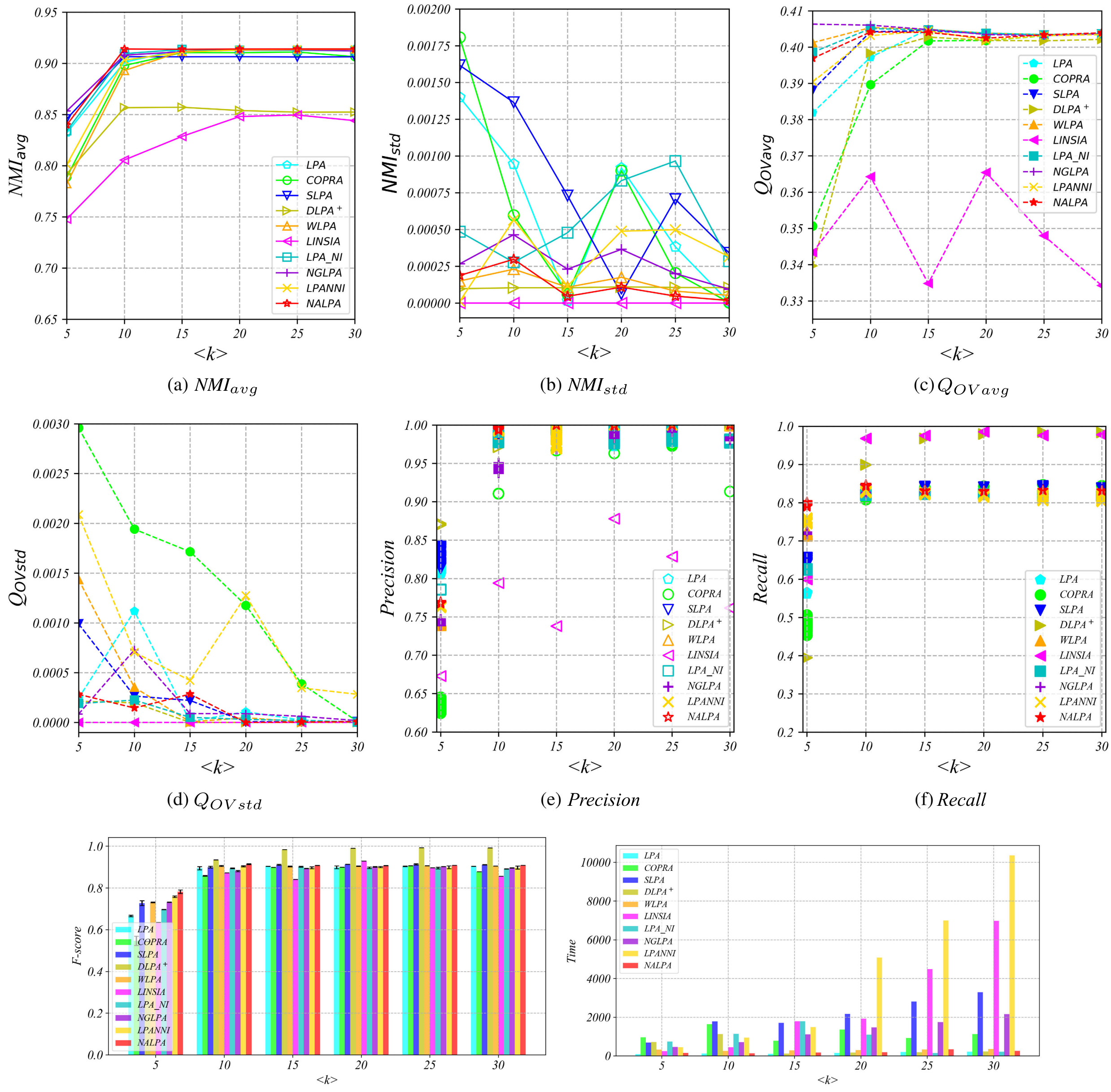


FIGURE 8. Results for the synthetic networks with overlapping communities with O_n varying when $N = 10000$.

nodes by node gravitation and node importance inspired from LeaderRank [26]. LPANNI uses the influence of neighbor nodes to identify the ownership of nodes. NALPA takes node abilities to attract more nodes and make them launch more labels and uses label influence to accept crucial labels for enhancing the model ability of distinguishing different community nodes, thus, it achieves better performance than most methods. On the contrary, DLPA$^+$ obtains poorer results than others except LINSIA because its inflation factor cannot be adaptive to control overlapping rate well in large networks, and LINSIA obtains the worst result because all labels with the largest importance are spread to other nodes.

In terms of Q_{OV}, most algorithms also behave similarly as shown in Figures 6(e) and 6(f), but the results of LPA and COPRA are away from the first tier. NALPA obtains good performance because attraction ability can draw more nodes to spread labels and increase the dense degree among nodes to promote the closeness of community, then its Q_{OV} is higher than most methods. On the other hand, these results decline more obviously than those of *NMI* when μ increases



(a) NMI_{avg} (b) NMI_{std} (c) Q_{OVavg}

(d) Q_{OVstd} (e) *Precision* (f) *Recall*

(g) *F-score* (h) *Time*

FIGURE 9. Results for the synthetic networks with overlapping communities with <k> varying when N = 10000.

from 0.1 to 0.3, which shows Q_{OV} is more sensitive to mixing parameter μ than *NMI*.

As for *Precision*, as shown in Figures 6(i) and 6(j), the algorithms except COPRA, LINSIA and NGLPA behave similarly. SLPA, LPANNI and NALPA gain better results than other methods. LPA, DLPA$^+$, WLPA and LPA_NI obtain the results which are slightly worse than SLPA, LPANNI and NALPA. We can find that the *Precision* of COPRA shows an oscillation trend, and the stability of COPRA and NGLPA is worse than other methods. In terms of *Recall*, LINSIA ranks the first and DLPA$^+$ gains suboptimal *Recall* when $\mu = 0.1$ as shown in Figure 6(k). When $\mu = 0.3$, as shown in Figure 6(l), LINSIA also achieves the best results and the *Recall* of DLPA$^+$ declines and is closed to most methods. As for *F-score*, as shown in Figure 6(m), DLPA$^+$ and LINSIA ranks the first and the second when $\mu = 0.1$, respectively. LPANNI and NALPA get better results than other methods except DLPA$^+$ and LINSIA. When $\mu = 0.3$, DLPA$^+$ and LINSIA also keep high *F-score*, and NALPA ranks the first with some O_m values. It can be seen that LINSIA has low *Precision* and high *Recall*, and its *Precision* is higher than those for the synthetic networks without overlapping communities, then it gets better *F-score* than most methods for alleviating over-detection problem.

TABLE 10. Average rank of algorithms for the synthetic networks with overlapping communities.

Criteria	Network	LPA	COPRA	SLPA	DLPA$^+$	WLPA	LINSIA	LPA_NI	NGLPA	LPANNI	NALPA
NMI_{avg}	LFR3-16	5.35	3.71	5.14	5.85	6.21	10.00	5.92	7.28	3.21	**2.28**
	LFR19-32	5.42	7.85	4.14	9.00	3.21	10.00	5.42	5.92	2.57	**1.42**
NMI_{std}	LFR3-16	8.21	9.00	8.35	5.85	4.42	**1.00**	6.71	5.57	3.50	2.21
	LFR19-32	7.42	7.14	6.42	4.92	6.78	**1.00**	6.14	7.78	4.50	2.85
Q_{OVavg}	LFR3-16	7.78	8.14	5.57	2.28	8.07	10.00	3.57	2.64	4.78	**2.14**
	LFR19-32	8.71	8.00	5.50	7.42	2.28	9.85	3.92	**1.57**	5.14	2.71
Q_{OVstd}	LFR3-16	9.21	9.64	6.92	5.71	3.50	**1.00**	6.64	4.57	4.85	1.78
	LFR19-32	8.28	9.14	5.85	5.00	5.14	**1.00**	5.64	5.85	5.71	3.35
P_{avg}	LFR3-16	5.35	9.35	3.78	6.92	2.71	9.64	5.57	7.92	2.21	**1.50**
	LFR19-32	5.00	9.07	3.42	7.42	3.21	9.92	5.92	7.71	**1.28**	2.14
P_{std}	LFR3-16	8.00	6.42	7.07	3.00	6.35	**1.00**	6.28	8.64	4.50	3.21
	LFR19-32	7.21	7.71	6.78	3.07	6.42	**1.00**	6.28	9.28	3.78	3.42
R_{avg}	LFR3-16	6.42	9.50	6.42	2.21	8.42	**1.00**	7.71	3.35	6.07	3.85
	LFR19-32	7.00	9.50	6.00	2.35	8.21	**1.00**	8.07	3.64	5.71	3.50
R_{std}	LFR3-16	6.57	6.64	7.07	4.35	7.92	**1.00**	5.50	5.50	6.64	3.71
	LFR19-32	7.14	6.00	5.78	5.28	8.92	**1.00**	5.85	5.21	7.00	2.78
F_{avg}	LFR3-16	5.85	8.85	4.64	2.64	7.42	3.71	7.50	7.78	4.28	**2.28**
	LFR19-32	6.14	9.92	5.07	2.71	6.92	3.14	7.64	7.14	3.64	**2.64**
F_{std}	LFR3-16	7.07	6.00	6.21	3.71	7.71	**1.00**	5.64	8.07	5.64	3.92
	LFR19-32	6.71	7.50	5.78	4.21	7.64	**1.00**	5.71	8.35	5.42	2.64
Time(s)	LFR3-16	1.78	4.78	10.00	8.14	3.21	4.28	7.14	6.28	8.14	**1.21**
	LFR19-32	**1.00**	9.85	8.92	7.92	3.42	3.85	7.28	4.85	5.78	2.07

The *Precision* and *Recall* of DLPA$^+$, LPANNI and NALPA are high and balanced, they obtain good *F-score*. The *Precision* and *Recall* of COPRA is relatively balanced, but neither of the two metrics is high, so its *F-score* is not high. Although DLPA$^+$ and LINSIA cannot obtain good *NMI* and Q_{OV}, they have high *F-score*, which means they can detect accurate node pair in true communities in these networks.

As for *RT*, LPA and NALPA respectively obtain the best efficiency in different networks as shown in Figures 6(o) and 6(p). NALPA gets the best average rank and LPA ranks the second as shown in Table 10. For NALPA, the new label propagation mechanism acquires more nodes to spread multiple labels and makes nodes accept more labels in updating process. Meanwhile, label influence can reflect the accurate weight of labels when they arrive at other nodes. Then nodes can accept influential labels to converge rapidly. Among other baselines, COPRA renews node labels in the tth iteration according to the ones in the (t-1)th iteration, then it needs more running time. SLPA has the worst efficiency because it requires 100 iterations. The efficiency of DLPA$^+$ is poor in these networks, since it inflation factor cannot adjust the overlapping rate of nodes well in large networks, then it needs more iterations. WLPA obtains suboptimal efficiency because it simply updates the labels of nodes by label weight. With the increase of O_m when $\mu = 0.1$, the running time of LINSIA shows the trend of gradually rising, since LINSIA spreads all labels with the largest importance to other nodes that needs more iterations to converge. The efficiency of LPA_NI is better than SLPA and DLPA$^+$, however, it is worse than other methods for calculating node importance and label influence by Bayesian network from expert knowledge. NGLPA and LPANNI get worse efficiency than most methods, in which the former computes node importance by LeaderRank for initialization that needs to iteratively calculate the relations among nodes and their neighbors, and the latter considers more local topology information to increase the accuracy of node similarity measurement.

As for stability, we can find that LINSIA and NALPA obtain the best and suboptimal stability, respectively. LINSIA propagates all labels with the largest importance of nodes to avoid randomness and NALPA updates the labels of nodes in the ascending order of propagation ability, launches labels in the descending order of belonging coefficient and accepts labels in the descending order of label influence. On the contrary, the stability of LPA and COPRA is worse than other methods, since they randomly choose nodes to update labels.

For the synthetic networks with overlapping communities when $N = 10000$, the overall results of five metrics (*NMI*, Q_{OV}, *Precision*, *Recall* and *F-score*) of all methods are similar with those of synthetic networks when $N = 5000$ and do not decline obviously, which shows that they are not sensitive to network scale. Different with the results when $N = 5000$, the running time of these methods for the networks when $N = 10000$ increases because network scale has doubled. We also can find that NALPA has similar efficiency with LPA, which is better than other methods. Specifically, the running time of COPRA rises more than other methods, which increases when $O_m \in [2, 5]$ and declines or keeps steady when $O_m \in [5, 8]$, thus, COPRA's running time is sensitive to network scale. When $\mu = 0.3$, we can find that the running time of COPRA falls at $O_m = 6$, which shows it converges fast under $O_m = 6$. According to the average rank as shown in Table 10, NALPA can detect more satisfying and stable communities with high efficiency than most state-of-the-art methods.

For the synthetic networks with overlapping communities when O_n varies, as shown in Figure 8, the five evaluations decline with the increase of O_n, which illustrates that the number of overlapping nodes has an influence on the community results. With the increase of overlapping nodes, networks become more intricate, then community detection methods cannot discover accurate communities. On the contrary, the *Precision* of LINSIA rises with the increase of O_n. LINSIA can detect hubs and outliers [17] that increases the accuracy of detecting the node pairs in the same community. As for *RT*, COPRA needs more time with the increase of O_n, since it requires more iterations to detect overlapping nodes. The efficiency of other methods keeps steady.

For the synthetic networks with overlapping communities when $<k>$ varies, as shown in Figure 9, the five evaluations of all methods at $<k> = 5$ are lower than those of methods at other $<k>$ values, which denotes that community detection methods cannot achieve good performance in too sparse synthetic networks. With the increase of $<k>$, the standard deviation of COPRA declines, which means its stability becomes better. LINSIA gets the worst and undulant Q_{OV} and *Precision* but the best *Recall* for over-detection problem. Meanwhile, when $<k>$ is greater than 10, these metrics keep steady with the increase of $<k>$. In terms of *RT*, the efficiency of SLPA, LINSIA and LPANNI rises with the increase of $<k>$, especially, LPANNI needs more running time in large average degree networks, since it computes the similarity between node v_i and its direct or indirect neighbors whose distances to node v_i are less than $\alpha = 3$. Thus, with the increase of $<k>$, the number of the indirect neighbors of node v_i shows an exponential growth trend, and the running time of LPANNI increases fast.

B. RESULTS FOR REAL-WORLD NETWORKS

The experimental results for the real-world networks with known and unknown communities are shown in Tables 11 and 12, respectively.

In terms of *NMI*, NALPA obtains better results than other methods as shown in Table 11. It is worth noting that, for Karate and Dolphin networks, the difference between NALPA and baselines is significant. In small-scale networks, the number of nodes is less. Then the number of initial unique labels is relatively small, and alterable label influence may reflect the difference among nodes well, therefore, the results are quite different. In terms of Q_{OV}, LPANNI gets the best average rank. We can find that NALPA provides the highest *NMI* with the middle Q_{OV} while DLPA$^+$ obtains the highest Q_{OV} with the middle *NMI* for Polbook network, which shows that the maximum Q_{OV} may not lead to the best partition of networks in some cases in conformity with the conclusion reported in [8]. With the increase of network scale, the *NMI* and Q_{OV} of LINSIA become poorly for the reasons mentioned above. The *NMI* of DLPA$^+$ becomes better with the increase of network scale, while its Q_{OV} becomes weaker. As for *Precision*, *Recall* and *F-score*, DLPA$^+$ ranks the first in terms of *Precision*, however, it gets the worst *Recall*, then it gets poor *F-score*. We consider that network scale also has an influence on the *Recall* of DLPA$^+$. Analyzing these real-world networks along with synthetic networks, we can find that the *Recall* of DLPA$^+$ rises with the increase of network scale, then its *F-score* increases with increasing network scale. On the other hand, LINSIA obtains the best *Recall* but the worst *Precision* due to the problem of over-detection, thus, it gets poor *F-score*. On the whole, although NALPA gets poor *Precision* and *Recall* in Football network, NALPA gains the best average rank in terms of *F-score* for balanced and good *Precision* and *Recall*. In terms of *RT*, NALPA has higher efficiency than other methods except LPA because the new label propagation mechanism is effective.

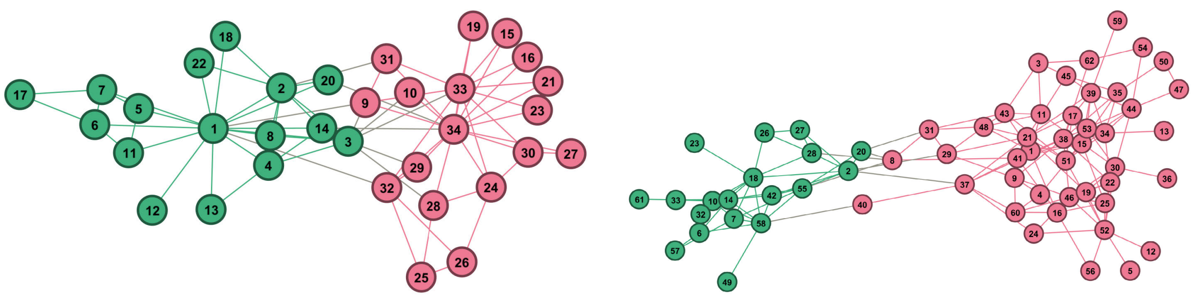
As shown in Table 12, NALPA obtains the best average rank in terms of Q_{OV}, although it obtains middle Q_{OV} in Jazz, Email and Enron networks. In the three networks, the difference with the result of NALPA and the optimal value is small. The attraction ability can attract more nodes to launch labels and increases the tightness of nodes in communities, however, this way may not fit the relative dense networks, such as Email network ($M = 5451$ and $dia = 8$). NALPA obtains the best Q_{OV} in the largest network Amazon, and LPANNI and LPA_NI ranks the second and the third, respectively. We can find that NALPA can obtain good results in large-scale networks. NALPA acquires similar efficiency with LPA, which illustrates that NALPA owns near linear complexity and good efficiency.

As for stability, we can find that the results of these methods in large-scale networks are more stable than those in small-scale networks comparing with the results in Tables 11 and 12, in other words, network scale has an influence on the stability of the algorithms. In small-scale networks, label propagation converges fast for relatively simple network structure, thus, the results of unstable algorithms are different and exist relatively large fluctuation. NALPA achieves the smallest standard deviation in most networks neglecting LINSIA that means the results of NALPA are more stable than other methods, since NALPA updates nodes and chooses labels according to the regulation order to lighten the instability. Among other label propagation algorithms, DLPA$^+$ comes to the third, WLPA, LPA_NI and LPANNI are slightly worse, however, the stability of LPA, COPRA, SLPA and NGLPA is poor.

In summary, NALPA obtains better and more stable results than state-of-the-art approaches in most cases, as the performance and efficiency of NALPA are improved by the new label propagation mechanism with node abilities and label influence.

TABLE 11. Results for real-world networks with known communities.

Criterion	Network	LPA	COPRA	SLPA	DLPA$^+$	WLPA	LINSIA	LPA_NI	NGLPA	LPANNI	NALPA
NMI_{avg}	Karate	0.4697 (8)	0.3596 (10)	0.6915 (3)	0.5489 (6)	0.5016 (7)	0.5992 (5)	0.6598 (4)	0.4408 (9)	0.7782 (2)	**1.0000 (1)**
	Dolphin	0.6581 (5)	0.5976 (8)	0.6678 (4)	0.4753 (10)	0.6599 (6)	0.7138 (2)	0.6436 (7)	0.7108 (3)	0.5809 (9)	**0.8167 (1)**
	Polbook	0.5583 (3)	0.5556 (5)	0.5574 (4)	0.4993 (9)	0.5709 (2)	0.4833 (10)	0.5409 (6)	0.5353 (7)	0.5207 (8)	**0.5713 (1)**
	Football	0.8521 (9)	0.8836 (7)	0.8862 (6)	0.9044 (2)	0.9013 (3)	0.7423 (10)	0.8823 (8)	0.8887 (5)	0.8997 (4)	**0.9068 (1)**
	Average Rank	6.25	7.50	4.25	6.75	4.50	6.75	6.25	6.00	5.75	**1.00**
NMI_{std}	Karate	0.2342 (10)	0.0955 (5)	0.2184 (9)	**0.0000 (1)**	0.1427 (8)	**0.0000 (1)**	0.1192 (7)	0.0969 (6)	0.0673 (4)	**0.0000 (1)**
	Dolphin	0.1321 (8)	0.0987 (6)	0.1372 (9)	**0.0000 (1)**	0.0666 (5)	**0.0000 (1)**	0.1389 (10)	0.1147 (7)	0.0209 (4)	**0.0000 (1)**
	Polbook	0.0215 (5)	0.0343 (9)	0.0394 (10)	0.0082 (3)	0.0217 (6)	**0.0000 (1)**	0.0278 (8)	0.0232 (7)	0.0205 (4)	**0.0000 (1)**
	Football	0.0336 (10)	0.0244 (8)	0.0178 (6)	0.0098 (3)	0.0180 (7)	**0.0000 (1)**	0.0272 (9)	0.0121 (4)	0.0132 (5)	0.0096 (2)
	Average Rank	8.25	7.00	8.50	2.00	6.50	**1.00**	8.50	6.00	4.25	1.25
$Q_{OV\,avg}$	Karate	0.3082 (9)	0.2348 (10)	0.3742 (6)	0.4210 (2)	0.3682 (7)	0.3989 (5)	0.4136 (4)	0.3314 (8)	0.4147 (3)	**0.4213 (1)**
	Dolphin	0.4968 (5)	0.3741 (9)	0.4757 (6)	0.5166 (3)	0.3695 (10)	0.3878 (8)	0.5055 (4)	0.5189 (2)	**0.5423 (1)**	0.4006 (7)
	Polbook	0.4968 (5)	0.4884 (7)	0.4923 (6)	**0.5249 (1)**	0.5070 (3)	0.4521 (10)	0.5048 (4)	0.4600 (9)	0.5097 (2)	0.4627 (8)
	Football	0.5922 (8)	0.5972 (6)	0.6016 (3)	0.5960 (7)	0.5981 (5)	0.5853 (9)	0.5985 (4)	0.5848 (10)	**0.6090 (1)**	0.6017 (2)
	Average Rank	6.75	8.00	5.25	3.25	6.25	8.00	4.00	7.25	**1.75**	4.50
$Q_{OV\,std}$	Karate	0.1207 (10)	0.1018 (9)	0.0868 (8)	**0.0000 (1)**	0.0817 (7)	**0.0000 (1)**	0.0659 (6)	0.0366 (5)	0.0144 (4)	**0.0000 (1)**
	Dolphin	0.0603 (10)	0.0394 (6)	0.0554 (9)	**0.0000 (1)**	0.0251 (5)	**0.0000 (1)**	0.0486 (7)	0.0493 (8)	0.0124 (4)	**0.0000 (1)**
	Polbook	0.0144 (7)	0.0321 (10)	0.0213 (9)	0.0041 (3)	0.0062 (4)	**0.0000 (1)**	0.0202 (8)	0.0102 (6)	0.0063 (5)	**0.0000 (1)**
	Football	0.0171 (8)	0.0211 (10)	0.0107 (5)	0.0093 (4)	0.0137 (7)	**0.0000 (1)**	0.0130 (6)	0.0179 (9)	0.0067 (3)	0.0013 (2)
	Average Rank	8.75	8.75	7.75	2.25	5.75	**1.00**	6.75	7.00	4.00	1.25
P_{avg}	Karate	0.7301 (9)	0.7557 (8)	0.8847 (5)	0.9215 (3)	0.7940 (7)	0.8425 (6)	0.9115 (4)	0.6571 (10)	0.9480 (2)	**1.0000 (1)**
	Dolphin	0.9629 (4)	0.9604 (6)	0.9637 (3)	0.9547 (7)	0.8847 (10)	0.9268 (9)	0.9689 (2)	**1.0000 (1)**	0.9618 (5)	0.9372 (8)
	Polbook	0.7665 (6)	0.7768 (5)	0.7804 (4)	**0.8174 (1)**	0.7911 (2)	0.6439 (10)	0.7575 (8)	0.7656 (7)	0.7831 (3)	0.7147 (9)
	Football	0.6003 (8)	0.7225 (6)	0.7459 (5)	**0.8802 (1)**	0.7975 (2)	0.0930 (10)	0.5483 (9)	0.7894 (3)	0.7796 (4)	0.6315 (7)
	Average Rank	6.75	6.25	4.25	**3.00**	5.25	8.75	5.75	5.25	3.50	6.25
P_{std}	Karate	0.1979 (10)	0.0770 (6)	0.1408 (9)	**0.0000 (1)**	0.0445 (5)	**0.0000 (1)**	0.0900 (8)	0.0802 (7)	0.0036 (4)	**0.0000 (1)**
	Dolphin	0.0281 (7)	0.0287 (8)	0.0234 (6)	**0.0000 (1)**	0.0345 (10)	**0.0000 (1)**	0.0287 (9)	0.0000 (1)	0.0023 (5)	0.0005 (4)
	Polbook	0.0256 (7)	0.0491 (10)	0.0332 (8)	0.0058 (3)	0.0175 (6)	**0.0000 (1)**	0.0367 (9)	0.0138 (4)	0.0162 (5)	**0.0000 (1)**
	Football	0.1098 (9)	0.0759 (6)	0.0845 (8)	0.0007 (2)	0.0840 (7)	**0.0000 (1)**	0.1225 (10)	0.0506 (4)	0.0513 (5)	0.0043 (3)
	Average Rank	8.25	7.50	7.75	1.75	7.00	**1.00**	8.75	4.00	4.75	2.25
R_{avg}	Karate	0.7794 (6)	0.6918 (9)	0.9343 (4)	0.5164 (10)	0.9897 (3)	**1.0000 (1)**	0.7243 (8)	0.7356 (7)	0.9073 (5)	**1.0000 (1)**
	Dolphin	0.6035 (5)	0.6860 (4)	0.5966 (6)	0.3615 (10)	0.9669 (2)	**1.0000 (1)**	0.5723 (8)	0.5783 (7)	0.4800 (9)	0.9655 (3)
	Polbook	0.8243 (4)	0.8111 (6)	0.8189 (5)	0.6601 (10)	0.8022 (7)	**0.9156 (1)**	0.7918 (8)	0.8884 (3)	0.7854 (9)	0.8980 (2)
	Football	0.8912 (5)	0.8847 (6)	0.9058 (3)	0.8429 (9)	0.8678 (7)	**1.0000 (1)**	0.9095 (2)	0.8270 (10)	0.9031 (4)	0.8576 (8)
	Average Rank	5.00	6.25	4.50	9.75	4.75	**1.00**	6.50	6.75	6.75	3.50
R_{std}	Karate	0.0183 (4)	0.0360 (6)	0.0775 (9)	**0.0000 (1)**	0.0305 (5)	**0.0000 (1)**	0.0690 (8)	0.0412 (7)	0.1004 (10)	**0.0000 (1)**
	Dolphin	0.2087 (9)	0.2486 (10)	0.1797 (8)	**0.0000 (1)**	0.0900 (5)	**0.0000 (1)**	0.1726 (7)	0.1683 (6)	0.0439 (3)	0.0833 (4)
	Polbook	0.0539 (5)	0.0976 (10)	0.0697 (7)	0.0389 (4)	0.0817 (8)	**0.0000 (1)**	0.0827 (9)	0.0540 (6)	0.0341 (3)	**0.0000 (1)**
	Football	0.0506 (7)	0.0445 (6)	0.0197 (2)	0.0278 (5)	0.0510 (8)	**0.0000 (1)**	0.0224 (3)	0.0569 (10)	0.0264 (4)	0.0534 (9)
	Average Rank	6.25	8.00	6.50	2.75	6.50	**1.00**	6.75	7.25	5.00	3.75
F_{avg}	Karate	0.7372 (7)	0.6992 (8)	0.9038 (4)	0.6619 (10)	0.8801 (5)	0.9145 (3)	0.8008 (6)	0.6934 (9)	0.9241 (2)	**1.0000 (1)**
	Dolphin	0.7242 (5)	0.7743 (4)	0.7232 (6)	0.5244 (10)	0.9220 (3)	**0.9620 (1)**	0.7064 (8)	0.7204 (7)	0.6394 (9)	0.9493 (2)
	Polbook	0.7929 (5)	0.7873 (6)	0.7965 (2)	0.7299 (10)	0.7935 (4)	0.7561 (9)	0.7707 (8)	**0.8213 (1)**	0.7835 (7)	0.7959 (3)
	Football	0.7104 (8)	0.7926 (6)	0.8153 (4)	**0.8609 (1)**	0.8281 (3)	0.1702 (10)	0.6756 (9)	0.8055 (5)	0.8358 (2)	0.7229 (7)
	Average Rank	6.25	6.00	4.00	7.75	3.75	5.75	7.75	5.50	5.00	**3.25**
F_{std}	Karate	0.1003 (8)	0.1447 (10)	0.1067 (9)	**0.0000 (1)**	0.0293 (4)	**0.0000 (1)**	0.0503 (5)	0.0626 (7)	0.0513 (6)	**0.0000 (1)**
	Dolphin	0.1468 (9)	0.1831 (10)	0.1285 (7)	**0.0000 (1)**	0.0559 (5)	**0.0000 (1)**	0.1311 (8)	0.1159 (6)	0.0344 (3)	0.0493 (4)
	Polbook	0.0276 (6)	0.0456 (10)	0.0302 (7)	0.0269 (5)	0.0400 (9)	**0.0000 (1)**	0.0378 (8)	0.0218 (4)	0.0162 (3)	**0.0000 (1)**
	Football	0.0786 (9)	0.0513 (5)	0.0559 (7)	0.0136 (2)	0.0522 (6)	**0.0000 (1)**	0.0953 (10)	0.0329 (3)	0.0339 (4)	0.0692 (8)
	Average Rank	8.00	8.75	7.50	2.25	6.00	**1.00**	7.75	5.00	4.00	3.50
Time(s)	Karate	0.0080 (3)	0.1788 (8)	6.4314 (10)	0.0071 (1)	0.1219 (7)	0.1101 (6)	0.1024 (5)	0.1788 (8)	0.0085 (4)	**0.0071 (1)**
	Dolphin	**0.2475 (1)**	0.4777 (8)	13.3722 (10)	0.3522 (4)	0.4155 (7)	0.5848 (9)	0.3485 (3)	0.3527 (5)	0.3905 (6)	0.3433 (2)
	Polbook	0.5751 (2)	1.0289 (6)	35.3845 (10)	0.8963 (4)	0.8330 (3)	1.4071 (9)	1.1117 (7)	1.0010 (5)	1.1159 (8)	**0.4562 (1)**
	Football	**0.5068 (1)**	1.0073 (6)	48.5151 (10)	0.6431 (4)	0.5485 (3)	2.2475 (8)	0.6656 (5)	13.9052 (9)	1.2825 (7)	0.5436 (2)
	Average Rank	1.75	7.00	10.00	3.25	5.00	8.00	5.00	6.75	6.25	**1.50**



(a) Karate (b) Dolphin

FIGURE 10. Community results of some networks for NALPA.

We present the community results of our method for Karate and Dolphin networks, as shown in Figure 10. In particular, the experimental results in some networks (e.g., Dolphin and Polbook networks) show that the maximum *NMI* does not correspond to the best Q_{OV}, and vice versa. Given an example of Polbook network to explain the reason. By analyzing the true communities of Polbook network, we can find that Polbook network owns three community as shown

TABLE 12. Results for real-world networks without known communities.

Criterion	Network	LPA	COPRA	SLPA	DLPA+	WLPA	LINSIA	LPA_NI	NGLPA	LPANNI	NALPA
Q_{OVavg}	Lesmis	0.5142 (8)	0.4846 (9)	0.5397 (4)	0.5446 (3)	0.5515 (2)	0.5396 (5)	0.5262 (7)	0.4477 (10)	0.5342 (6)	**0.5552 (1)**
	Jazz	0.3457 (5)	0.3954 (3)	0.3457 (5)	0.2821 (8)	0.3624 (4)	0.2178 (10)	**0.4343 (1)**	0.2812 (9)	0.3136 (7)	0.4322 (2)
	Email	0.2909 (2)	**0.2915 (1)**	0.2905 (5)	0.2401 (10)	0.2908 (4)	0.2503 (9)	0.2726 (7)	0.2734 (6)	0.2909 (2)	0.2504 (8)
	Netscience	0.9004 (7)	0.8784 (8)	0.9043 (6)	0.8456 (9)	0.9279 (2)	0.6018 (10)	0.9140 (4)	0.9209 (3)	0.9070 (5)	**0.9283 (1)**
	Power	0.6801 (5)	0.1696 (10)	0.6225 (8)	0.5893 (9)	0.7731 (2)	0.6365 (7)	0.7473 (4)	0.7631 (3)	0.6608 (6)	**0.7818 (1)**
	PGP	0.7661 (4)	0.5117 (10)	0.7641 (5)	0.6761 (8)	0.6231 (9)	0.6948 (7)	0.7861 (3)	0.8092 (2)	0.7575 (6)	**0.8234 (1)**
	Cond2003	0.6306 (5)	0.6306 (5)	0.6341 (2)	0.4764 (10)	0.5959 (7)	0.4816 (9)	0.6313 (3)	0.5907 (8)	0.6312 (4)	**0.6777 (1)**
	Enron	0.3163 (4)	0.3068 (7)	0.3140 (5)	0.2844 (9)	**0.3191 (1)**	0.2887 (8)	0.3182 (2)	0.2728 (10)	0.3178 (3)	0.3079 (6)
	Cond2005	0.5355 (6)	0.4256 (9)	0.6019 (5)	0.4371 (8)	0.6117 (3)	0.3296 (10)	0.6111 (4)	0.4431 (7)	0.6175 (2)	**0.6301 (1)**
	Amazon	0.7389 (7)	0.7699 (4)	0.7405 (6)	0.6415 (9)	0.7690 (5)	0.6237 (10)	0.7714 (3)	0.7137 (8)	0.7856 (2)	**0.7901 (1)**
	Average Rank	5.30	6.60	5.10	8.30	3.90	8.50	3.80	6.60	4.30	**2.30**
Q_{OVstd}	Lesmis	0.0697 (9)	0.0468 (7)	0.0365 (4)	0.0006 (2)	0.0402 (5)	**0.0000 (1)**	0.0627 (8)	0.0815 (10)	0.0430 (6)	0.0006 (2)
	Jazz	0.0895 (9)	0.0713 (7)	0.0916 (10)	**0.0000 (1)**	0.0804 (8)	**0.0000 (1)**	0.0363 (4)	0.0463 (5)	0.0581 (6)	**0.0000 (1)**
	Email	0.0198 (7)	0.0256 (10)	0.0072 (4)	0.0022 (3)	0.0242 (9)	**0.0000 (1)**	0.0191 (6)	0.0215 (8)	0.0079 (5)	**0.0000 (1)**
	Netscience	0.0038 (7)	0.0184 (10)	0.0068 (9)	0.0032 (6)	0.0065 (8)	**0.0000 (1)**	0.0023 (3)	0.0026 (4)	0.0026 (4)	0.0002 (2)
	Power	0.0019 (4)	0.0089 (10)	0.0039 (9)	0.0017 (3)	0.0037 (7)	**0.0000 (1)**	0.0034 (6)	0.0019 (4)	0.0038 (8)	0.0008 (2)
	PGP	0.0066 (8)	0.0955 (10)	0.0059 (7)	0.0014 (3)	0.0322 (9)	**0.0000 (1)**	0.0039 (5)	0.0020 (4)	0.0049 (6)	0.0001 (2)
	Cond2003	0.0030 (6)	0.0039 (8)	0.0029 (5)	0.0032 (7)	0.0055 (9)	**0.0000 (1)**	0.0014 (4)	0.0097 (10)	0.0008 (3)	0.0006 (2)
	Enron	0.0009 (8)	0.0007 (7)	0.0088 (10)	0.0002 (3)	0.0005 (5)	**0.0000 (1)**	0.0034 (9)	0.0001 (2)	0.0005 (5)	0.0002 (3)
	Cond2005	0.0064 (8)	0.0039 (5)	0.0027 (4)	0.0015 (2)	0.0194 (10)	**0.0000 (1)**	0.0086 (9)	0.0040 (6)	0.0016 (3)	0.0049 (7)
	Amazon	0.0083 (10)	0.0053 (8)	0.0022 (6)	0.0052 (7)	0.0008 (5)	**0.0000 (1)**	0.0005 (4)	0.0078 (9)	0.0004 (2)	0.0004 (2)
	Average Rank	7.60	8.20	6.80	3.70	7.50	**1.00**	5.80	6.20	4.80	2.40
Time(s)	Lesmis	**0.1925 (1)**	0.4285 (7)	19.5511 (10)	0.2447 (3)	0.3132 (4)	0.4528 (8)	0.4174 (6)	4.5464 (9)	0.3339 (5)	0.2024 (2)
	Jazz	2.2609 (2)	4.1365 (5)	201.9469 (10)	2.6987 (3)	4.4134 (6)	6.3256 (7)	3.3010 (4)	46.7322 (8)	63.4638 (9)	**2.0781 (1)**
	Email	15.4902 (2)	143.7178 (8)	415.8542 (10)	109.7681 (7)	22.2821 (3)	29.8843 (4)	72.2570 (5)	102.1098 (6)	305.5800 (9)	**8.2727 (1)**
	Netscience	4.1330 (2)	33.2918 (7)	245.5020 (10)	5.8815 (3)	23.5540 (6)	8.7249 (4)	17.440 (5)	73.9827 (9)	46.2049 (8)	**4.0007 (1)**
	Power	**13.2999 (1)**	341.5182 (10)	211.0747 (9)	167.6967 (8)	145.2718 (7)	24.3082 (2)	138.3604 (6)	92.9416 (4)	111.9992 (5)	51.4232 (3)
	PGP	**55.3982 (1)**	3909.1786 (10)	767.6998 (7)	518.1078 (5)	2034.5744 (9)	252.4939 (3)	558.5890 (6)	390.1165 (4)	1080.641 (8)	66.4761 (2)
	Cond2003	**555.6496 (1)**	10471.5065 (10)	4624.2670 (9)	3997.4615 (8)	3560.0126 (7)	2106.7835 (4)	2090.6047 (3)	2316.1850 (5)	3209.3535 (6)	747.0775 (2)
	Enron	**783.3653 (1)**	5893.1192 (8)	6197.1434 (9)	6283.5557 (10)	2563.4053 (3)	3694.3290 (6)	3112.7570 (4)	3666.8810 (5)	4357.6440 (7)	1076.3930 (2)
	Cond2005	3036.3413 (2)	24132.8245 (10)	6206.6146 (7)	6705.6510 (8)	4883.3493 (6)	10454.4580 (9)	3480.0526 (3)	3853.6380 (4)	4632.9150 (5)	**1104.0530 (1)**
	Amazon	14688.7725 (3)	377662.4635 (10)	360883.7190 (9)	35257.9160 (5)	**13168.8880 (1)**	144404.5910 (8)	22555.8957 (4)	38266.2185 (6)	55613.5816 (7)	13202.8785 (2)
	Average Rank	**1.60**	8.50	9.00	6.00	5.20	5.50	4.60	6.00	6.90	1.70

in Figure 11(a), the orange and light purple communities with majority nodes and the green community with scattered nodes, specially, some nodes (e.g., nodes 8, 29, 49 and 77) in the green community are not closely connected to other nodes. Meanwhile, the nodes in the green community locate in the border of the orange and light purple communities, which may be caused by the people with the neutral point of view in real world. On the contrary, the communities with the best Q_{OV} are some small communities where inside nodes are more close. Thus, the communities with the best *NMI* is not in accordance with those with the best Q_{OV}. The communities detected by these algorithms are shown in Figure 11 (the red box shows the error-prone areas of methods). LPA, WLPA and NGLPA find three communities. However, some nodes belonging to the true green community are assigned to the orange and light purple communities. COPRA, SLPA and LPANNI detect four communities in Polbook network, and most errors focus on the nodes belonging to the true green community. These nodes are assigned to the orange and light purple communities or form new communities, however, these communities are far away from true partitions. DLPA$^+$ discovers the most communities, in which the nodes have dense inner connection, so that it gets the highest Q_{OV} but poor *NMI*. We can find that some methods based on the weight of nodes and labels (LINSIA, LPA_NI and NALPA) tend to discover two communities in Polbook network, and NALPA finds most accurate nodes in communities, which are the same as true partitions, so it obtains the best *NMI*.

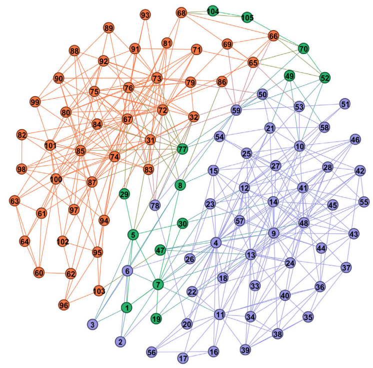
C. CASE STUDY

According to experimental results above, NALPA can provide good accuracy with the highest efficiency for detecting communities. Thus, we apply NALPA to discover drug communities in a TCM drug network where drugs are used for treating Chronic GlomeruloNephritis (CGN). In order to utilize NALPA to discover drug communities, we first select key word pairs "慢性肾小球肾炎 (chronic glomerulonephritis)" and "中医 (Chinese medicine)", and "慢性肾小球肾炎 (chronic glomerulonephritis)" and "中药 (Chinese native medicine)" to search literature in China National Knowledge Infrastructure (CNKI) and acquire 1126 literature, then use Chinese word embedding method [45] to encode words in literature to low-dimensional vectors, extract drugs and calculate their similarity to build a drug network, in which nodes represent drugs, and if two drugs are similar, they are linked to form an edge. Then we construct a drug network with 273 nodes and 520 edges, which is built based on drug similarity, then the drugs in the same community have similar efficacy and are signed for treating a class of symptoms of CGN. The results are shown in Figure 12.

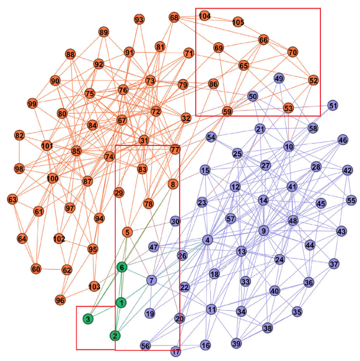
TABLE 13. Drugs with top-6 degree in communities.

Community	Chinese Name	English Name
purple	地黄	rehmannia root
	白芍	debark peony root
	麦冬	radix ophiopogonis
	陈皮	dried tangerine peel
	麻黄	ephedra
	肉桂	cinnamon
green	补骨脂	malaytea scurfpea fruit
	龙骨	drgonsbones
	杏仁	apricot seed
	桔梗	platycodon root
	赤小豆	rice bean
	巴戟天	morinda root

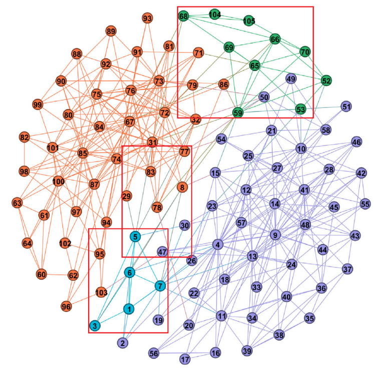
We choose two large drug communities (purple and green) to explain their rationality. The drugs with the top-6 degree in the two communities are shown in Table 13. In the purple community, the drugs are used for treating the syndromes of



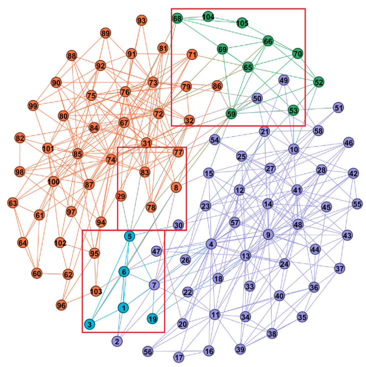
(a) true communities ($NMI = 1$)



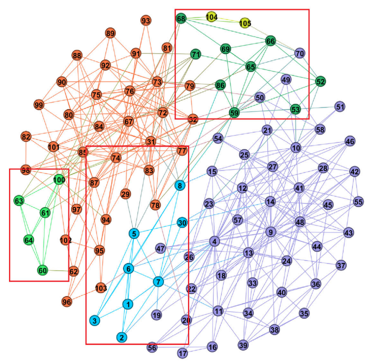
(b) communities detected by LPA



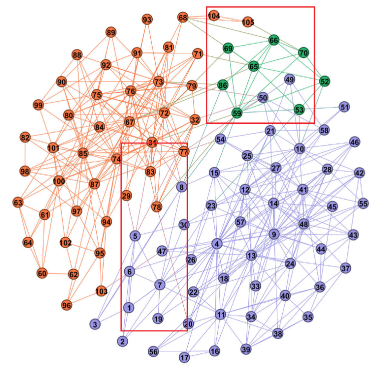
(c) communities detected by COPRA



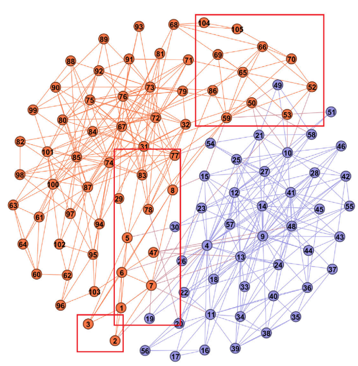
(d) communities detected by SLPA



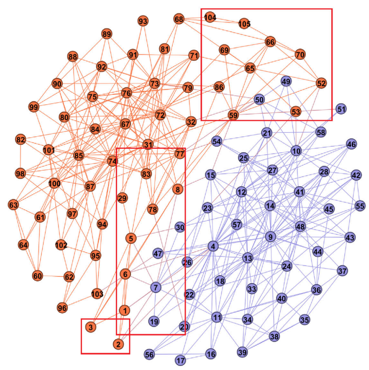
(e) communities detected by DLPA$^+$



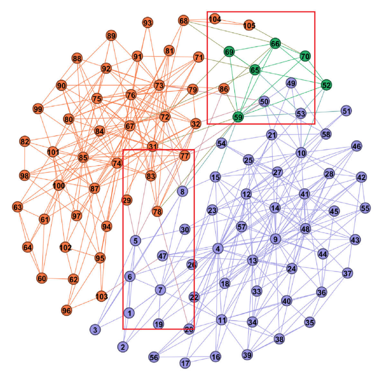
(f) communities detected by WLPA



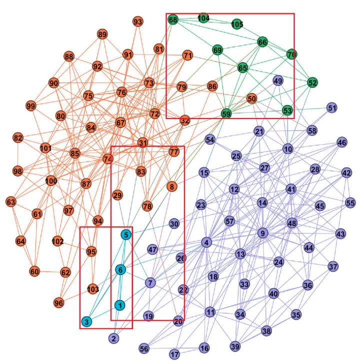
(g) communities detected by LINSIA



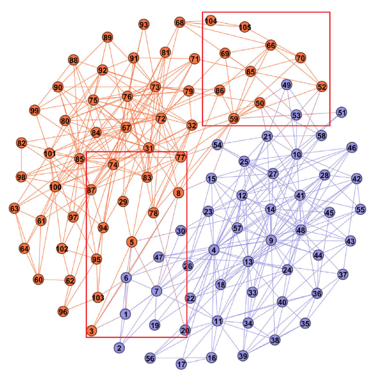
(h) communities detected by LPA_NI



(i) communities detected by NGLPA



(j) communities detected by LPANNI



(k) communities detected by NALPA

FIGURE 11. Community results for Polbook network.

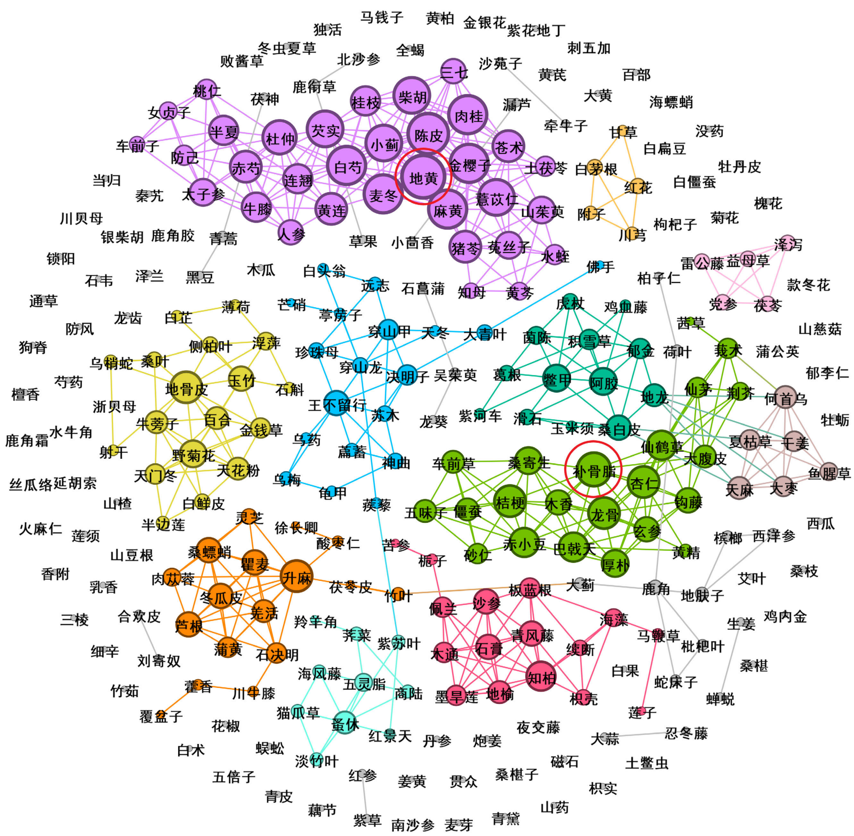


FIGURE 12. Drug communities for the drug network.

liver and kidney deficiency mainly according to the analysis of TCM doctors. "地黄 (rehmannia root)" with the largest degree in this community is the core drug for treating these syndromes, which is in accordance with [46]. In the green community, the drugs own the efficacy of nourishing liver and kidney. "补骨脂 (malaytea scurfpea fruit)" with the largest degree in this community is severed as the core drug with the efficacy in keeping with patents CN107875233-A and CN104491575-A. Thus, NALPA can detect accurate drug communities in drug networks.

VII. CONCLUSION

In this paper, a node ability based label propagation algorithm NALPA is proposed for community detection. We design four node abilities and label influence to provide a new label propagation mechanism and decide the order of label updating and label receiving to enhance the efficiency and handle the instability. We validate the effectiveness of NALPA on 42 synthetic and 14 real-world networks. In addition, NALPA is introduced to detect drug communities in a drug network and can find effective drug communities.

We also find that NALPA is inferior to some algorithms in some networks. Designing the elaborate strategies of acceptance ability for each node and adaptive parameters to detect more accurate communities is the important area of future research. It would also be interesting to combine deep learning and node abilities in community detection algorithms [47].

REFERENCES

[1] X. Wang, G. Liu, and J. Li, "Overlapping community detection based on structural centrality in complex networks," *IEEE Access*, vol. 5, pp. 25258–25269, 2017.

[2] S. Mishra, C. Hota, L. Kumar, and A. Nayak, "An evolutionary GA-based approach for community detection in IoT," *IEEE Access*, vol. 7, pp. 100512–100534, 2019.

[3] M. Tang, Q. Liu, T. Ma, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Dhelaan, "$\mathcal{K}$-lowest-influence overlapping nodes based community detection in complex networks," *IEEE Access*, vol. 7, pp. 109646–109661, 2019.

[4] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature Phys.*, vol. 6, no. 11, pp. 888–893, Aug. 2010.

[5] Y. Zhao, S. Li, and F. Jin, "Identification of influential nodes in social networks with community structure based on label propagation," *Neurocomputing*, vol. 210, pp. 34–44, Oct. 2016.

[6] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018.

[7] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, "LPANNI: Overlapping community detection using label propagation in large-scale complex networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1736–1749, Sep. 2019.

[8] Y. Zhang, Y. Liu, J. Li, J. Zhu, C. Yang, W. Yang, and C. Wen, "WOCDA: A whale optimization based community detection algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 539, Feb. 2020, Art. no. 122937.

[9] K. Berahmand and A. Bouyer, "LP-LPA: A link influence-based label propagation algorithm for discovering community structures in networks," *Int. J. Modern Phys. B*, vol. 32, no. 6, Feb. 2018, Art. no. 1850062.

[10] T. Chakraborty, S. Ghosh, and N. Park, "Ensemble-based overlapping community detection using disjoint community structures," *Knowl.-Based Syst.*, vol. 163, pp. 241–251, Jan. 2019.

[11] 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, 2014, Art. no. 056117.

[12] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[13] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103018.

[14] C. Tong, J. Niu, J. Wen, Z. Xie, and F. Peng, "Weighted label propagation algorithm for overlapping community detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 1–6.

[15] K. Liu, J. Huang, H. Sun, M. Wan, Y. Qi, and H. Li, "Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks," *Knowl.-Based Syst.*, vol. 89, pp. 487–496, Nov. 2015.

[16] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, and G. Sun, "A node influence based label propagation algorithm for community detection in networks," *Sci. World J.*, vol. 2014, Jun. 2014, Art. no. 627581.

[17] T. Wu, Y. Guo, L. Chen, and Y. Liu, "Integrated structure investigation in complex networks by label propagation," *Phys. A, Stat. Mech. Appl.*, vol. 448, pp. 68–80, Apr. 2016.

[18] X.-K. Zhang, J. Ren, C. Song, J. Jia, and Q. Zhang, "Label propagation algorithm for community detection based on node importance and label influence," *Phys. Lett. A*, vol. 381, no. 33, pp. 2691–2698, Sep. 2017.

[19] M. Shen and Z. Ma, "A novel node gravitation-based label propagation algorithm for community detection," *Int. J. Modern Phys. C*, vol. 30, no. 6, Jul. 2019, Art. no. 1950049.

[20] H.-L. Sun, J.-B. Huang, Y.-Q. Tian, Q.-B. Song, and H.-L. Liu, "Detecting overlapping communities in networks via dominant label propagation," *Chin. Phys. B*, vol. 24, no. 1, Jan. 2015, Art. no. 018703.

[21] Z. G. Zhang, *Social Psychology*, 2nd ed. Beijing, China: People's Education Press, 2008.

[22] C. Cao, Z. Z. Chen, J. Caverlee, L. A. Tang, C. Luo, and Z. C. Li, "Behavior-based community detection: Application to host assessment in enterprise information networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2018, pp. 1977–1985.

[23] L. F. Jing, *The Principle of Radar*, 5th ed. Beijing, China: Electronic Industry Press, 2015.

[24] Y. Zhang, Q. Y. Li, L. Huang, and J. Song, "Waveform design for joint radar-communication system with multi-user based on MIMO radar," in *Proc. IEEE Radar Conf.*, Jun. 2017, pp. 415–418.

[25] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 344–349.

[26] L. Lü, Y.-C. Zhang, C. H. Yeung, and T. Zhou, "Leaders in social networks, the delicious case," *PLoS ONE*, vol. 6, no. 6, Jun. 2011, Art. no. e21202.

[27] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.

[28] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropolog. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[29] D. Lusseau, "The emergent properties of a dolphin social network," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 270, no. S2, p. S186, Nov. 2003.

[30] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[31] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[32] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*. Reading, MA, USA: Addison-Wesley, 1993, pp. 41–43.

[33] P. M. Gleiser and L. Danon, "Community structure in jazz," *Adv. Complex Syst.*, vol. 6, no. 4, pp. 565–573, 2003.

[34] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 6, Dec. 2003, Art. no. 065103.

[35] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104.

[36] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[37] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of social networks based on social distance attachment," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 056122.

[38] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.

[39] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, Jan. 2009.

[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, Oct. 2013.

[41] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, Mar. 2009, Art. no. 033015.

[42] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, Jun. 2004, Art. no. 066133.

[43] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *J. Stat. Mech., Theory Exp.*, vol. 2009, no. 3, Mar. 2009, Art. no. P03024.

[44] M. Zhu, F. Meng, and Y. Zhou, "Semisupervised clustering for networks based on fast affinity propagation," *Math. Problems Eng.*, vol. 2013, pp. 1–13, 2013.

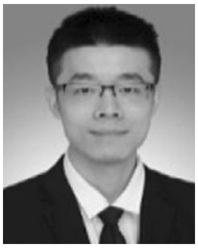
[45] Y. Zhang, Y. Liu, J. Zhu, Z. Zheng, X. Liu, W. Wang, Z. Chen, and S. Zhai, "Learning Chinese word embeddings from stroke, structure and pinyin of characters," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2019, pp. 1011–1020.

[46] H. Qiu, P. Fu, W. Fan, C. Zuo, P. Feng, P. Shi, L. Cao, F. Liu, L. Zhou, F. Chen, H. Zhong, Z. Gou, Y. Liang, and M. Shi, "Treatment of primary chronic glomerulonephritis with rehmannia glutinosa acteosides in combination with the angiotensin receptor blocker irbesartan: A randomized controlled trial," *Phytotherapy Res.*, vol. 28, no. 1, pp. 132–136, Mar. 2013.

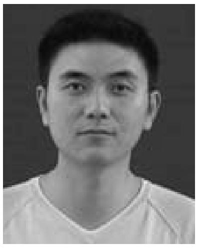
[47] L. Yang, X. C. Cao, D. X. He, C. Wang, and W. X. Zhang, "Modularity based community detection with deep learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 2252–2258.

YUN ZHANG received the B.S. degree from the Software School, in 2014, and the M.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests are complex network analysis, natural language processing, and TCM informatics.

YONGGUO LIU received the M.S. degree from Sichuan University, in 2000, and the Ph.D. degree in computer science from Chongqing University, in 2003. He finished his Postdoctoral Research at Shanghai Jiaotong University, in 2005. He joined the University of Electronic Science and Technology of China, in 2005, where he is currently a Full Professor with the School of Information and Software Engineering. His research interests include medical informatics and data mining.



JIAJING ZHU received the B.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests are TCM informatics, machine learning, and pattern recognition.



CHANGHONG YANG received the B.M. degree from the West China University of Medical Science, in 1993, and the Ph.D. degree in cartography and geographic information from the Chinese Academy of Science, in 2016. He is currently the Director and a Researcher of the Sichuan Center for Disease Control and Prevention, Institute for Public Health Information. His research interests include informatization, epidemiology, and environment and health.

WEN YANG received the M.S. degree in medicine from the Kunming Medical College, in 1990, supported by WHO, from 1994 to 1995, in Thailand and Australia for vector control. He joined the Sichuan Institute of Parasitic Diseases Control, in 1990. He is currently a Full Professor with the Sichuan Center for Disease Control and Prevention. His research interests include public health and infection diseases.

SHUANGQING ZHAI received the M.S. and Ph.D. degree from the Beijing University of Chinese Medicine, in 1988 and 2000, respectively. He is currently a famous TCM expert and has a long career in curing mental, centrum cerebrovascular, and digestive diseases. He is also an Academic Leader of the Nei Jing in State Administration of TCM. His research interests are TCM fundamental theories and Huang Di Nei Jing.