A Multi-Label Propagation Algorithm with the Double-Layer Filtering Strategy for Overlapping Community Detection

SHUYI YANG\textsuperscript{1,2} and JIA CAO\textsuperscript{1,2}

\textsuperscript{1}School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China
\textsuperscript{2}Engineering Research Center for Forestry-Oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China

Corresponding author: Jia Cao (e-mail: caojia@bjfu.edu.cn).

This work was supported by the National Natural Science Foundation of China under Grant 61602042.

ABSTRACT Overlapping community structures are universal in complex networks. Multi-label propagation approach, as a method for detecting overlapping community, have been received widespread attention for its near linear time complexity. In recent years, some algorithms based on multi-label propagation approach have been proposed, but there is still room to improve the accuracy of these algorithms to detect overlapping communities. In this paper, based on the research of previous multi-label propagation algorithms, a double-layer filtering strategy is proposed to help nodes filter labels and the node centrality of each node according to the number of its neighbors and the number of edges between its neighbors is used to fix the node update sequence. Then based on the double-layer filtering strategy, the new fixed node update sequence and the previous multi-label propagation algorithm, a new algorithm called MLPA-DF is proposed. Theory analysis show that MLPA-DF still has the near linear time complexity. And experimental results on synthetic and real-world networks show that MLPA-DF has better performance than previous multi-label propagation algorithms (COPRA, SLPA and LPANNI) in terms of the accuracy of detecting overlapping communities without reducing stability.

INDEX TERMS Complex network, community detection, overlapping community, label propagation

I. INTRODUCTION

Community structure is an important topological feature in the complex network. Usually, the nodes in the same community are tightly connected, while the nodes of different communities are sparsely connected [1]. Nodes belonging to one community are non-overlapping nodes, and nodes belonging to multiple communities are overlapping nodes. Communities can also be divided into overlapping and non-overlapping communities. There are only non-overlapping nodes in non-overlapping communities, while there are both overlapping and non-overlapping nodes in overlapping communities. Overlapping communities generally exist in complex networks.

Detecting overlapping communities is of great significance for complex networks. Many algorithms have been proposed to solve this problem. These algorithms can be divided into five main types [2]: The first type is the clique percolation algorithm (CPM), which is based on cliques to detect overlapping communities [3], [4]. The second type is link partitioning algorithm which focuses on edges instead of nodes in the networks [5], [6]. The third type utilizes local expansion and optimization functions to add nodes to different communities, such as OSLOM [7] and HSEI [8]. The fourth type is to use fuzzy detection algorithms (such as fuzzy c-means algorithm) [9] to detect overlapping communities. The last type of algorithm is based on multi-label propagation approach, such as COPRA [10]. This type of algorithm is an extension of the Label Propagation Algorithm (LPA) [11]. It assigns labels to nodes, allows nodes to continuously update labels according to some rules, and finally uses the labels of nodes to detect overlapping communities. Wu et al. call this type of algorithm multi-label propagation algorithm (MLPA) [12].

Nowadays, MLPA has gained popularity due to the near linear time complexity [13]. COPRA, the most well-known MLPA, has been deeply studied by researchers. Due to the problems caused by the randomness of COPRA’s label...
update strategy, some algorithms based on COPRA have been proposed [12], [14]-[18]. Compared with COPRA, these algorithms all improve the accuracy of detecting overlapping communities, but they have their own shortcomings. The stability of SLPA [14], DLPA [15] and LPA-E [16] is not good, the time complexity of BMLPA [12] and CLBLPA [17] increases, and the performance of LPANNI [18] on some small-scale networks is not ideal. Therefore, there is still room for improvement in this type of algorithm. In order to further improve the stability and accuracy of detecting overlapping communities, a novel MLPA is proposed in this paper, which is called MLPA-DF. MLPA-DF is an improved version of LPANNI, and it adopts the double-layer filtering strategy and a new fixed node update sequence. The main contributions of this paper are summarized as follows:

1) A double-layer filtering strategy is proposed to help each node select the appropriate labels and classify the selected labels into two categories.
2) A new fixed node update sequence is proposed based on the node centrality. The centrality of a node is determined by the number of its neighbors and the number of edges between its neighbors.
3) The double-layer filtering strategy replaces the label filtering method and dominant label in the label update strategy of LPANNI, and the new fixed node update sequence replaces the node update sequence of LPANNI. These changes allow the algorithm to achieve better accuracy and stability in detecting overlapping communities, and solve the problem of its poor performance on some networks.

The rest of this paper is organized as follows. Section II introduces some popular MLPAs. Section III presents the details of our proposed algorithm. Experimental results on synthetic and real-world networks are shown in Section IV. The conclusion is given in Section V.

II. RELATED WORK

MLPA is one of the effective methods to detect overlapping communities in complex networks and is the extension of LPA. In LPA and MLPA, labels are community identifiers. Each node is initially given a unique label, and updates labels according to the label update strategy. At the beginning, there are a lot of different labels in the network. As nodes are iteratively updated, closely connected nodes may have the same label, so the types of labels in the network are greatly reduced. After all nodes stop updating, the labels of the nodes determine the communities these nodes join. The difference between LPA and MLPA is mainly whether their label update strategies allow each node to have multiple labels. When a node updates the label, in LPA it selects only one label as its new label, while in MLPA it selects more than one label as its new label. That is, in MLPA, there are some nodes with multiple labels, these nodes are overlapping nodes, and they belong to multiple communities.

COPRA is the first MLPA. It improves the label update strategy of LPA and solves the problem that LPA cannot detect overlapping communities. And COPRA is the first to define the belonging coefficient to indicate the possibility that a node belongs to the community identified by a label. The main procedure of COPRA is as follow:

1) Initialization. Every node is given a unique label, and the belonging coefficient of the label is set to 1.
2) Updating labels. Repeatedly, all nodes update labels at the same time according to the labels of their neighbors, until the termination criterion proposed by Gregory [10] is satisfied. When a node updates labels, it first clears its current labels and receives all the labels of all neighbors. Then it recalculates the belonging coefficients of all labels. According to a global parameter $v$, the node selects labels whose belonging coefficients are larger than the threshold $1/v$ as its new labels and deletes other labels. If no label’s belonging coefficient is larger than $1/v$, the node randomly selects one of the labels with the largest belonging coefficient as its new label. Then normalizing the belonging coefficients of the new labels.
3) Detecting communities. All nodes with the same label join the same community. If a node has more than one label, it is an overlapping node and join more than one community. Communities that are totally contained by others are removed.

In sparse networks, the time complexity of COPRA is near $O(n)$.

Although COPRA can detect overlapping communities, the randomness of its label update strategy leads to its poor stability. Therefore, some algorithms have been proposed based on COPRA to improve both its accuracy and stability.

SLPA [14] uses the label update history to improve COPRA. In SLPA, every node has a memory for recording the label update history. The node updates the label, only one label is selected as the new label, and then record this label to its memory. If a label occurs more frequently in the memory of a node, it is more likely to be selected by neighbors of the node as the new label. After all node update several times, some labels are recorded in their memories. A global parameter $r$ is used as the threshold, and the memories delete the labels whose occurrence times are lower than $r$. The remaining labels determine which communities these nodes join. In sparse networks, the time complexity of SLPA is also near $O(n)$.

Each node is initially given a unique label in COPRA. BMLPA [12] and CLBLPA [17] improve the initialization method of COPRA. CLBLPA uses Leader Rank algorithm to select core nodes in the network, and only the core nodes are initially given a unique label [17]. BMLPA uses Rough core algorithm to classify nodes, and nodes in the same category are initially given the same label [12]. In addition, BMLPA uses balanced belonging coefficient instead of the belonging coefficient of COPRA to improve the accuracy of detecting overlapping communities. The time complexity of
BMLPA and CLBLPA in sparse networks are both $O(n \log n)$. Compared with COPRA, their time complexity has increased.

DLPA [15] and LPA-E [16] redefine the belonging coefficient and set the threshold for filtering labels in different ways from COPRA. In LPA-E and DLPA, information entropy and Jaccard index are used to calculate the belonging coefficients, respectively. And when the node $x$ updates labels in LPA-E and DLPA, their thresholds to help $x$ filter labels are both set to $1/|\text{Nb}(x)|$, where $|\text{Nb}(x)|$ is the number of $x$’s neighbors. Moreover, in DLPA, each node selects a label with the largest belonging coefficient from its own labels as the dominant label. When a node updates labels, only the dominant label of its neighbors can be received. The time complexity of LPA-E and DLPA in sparse networks are near $O(n)$.

LPANNI [18] uses a series of methods to improve COPRA. It has two phases. In the first phase, it calculates $NI$ (Node Importance) and $NNI$ (Neighbor Node Influence) and sorts all nodes in ascending order based on $NI$. The calculation method of $NI$ and $NNI$ can refer to Eq. (1) and Eq. (6) in reference [18]. The second phase is the label propagation which is similar to COPRA, including three steps:

1) Initialization. Each node is given a unique label and the label is the dominant label. The belonging coefficient of this label is set to 1.

2) Updating labels. Repeatedly, each node updates labels according to the ascending order of $NI$. When a node updates labels, it first clears its current labels and receives the dominant label of each neighbor. Then it recalculates the belonging coefficients of the received labels according to Eq. (8) and Eq. (9) in reference [18]. The node $x$ selects labels whose belonging coefficients are larger than the threshold $1/L_x(x)$ as its new labels and deletes other labels. Here $|L_x(x)|$ is the number of different labels received from neighbors by $x$. At the same time, it selects one of the labels with the largest belonging coefficient as its dominant label. If a label with largest belonging coefficient is the dominant label of the node in previous iteration, it will be selected as the dominant label first in this iteration. The termination criteria are as follows: the size of label set and the dominant labels of all nodes don’t change in two consecutive iterations, or the number of iterations reaches the upper limit.

3) Detecting communities. In this step, all labels are no longer divided into dominant labels and non-dominant labels. This step is the same as in COPRA.

LPANNI has good accuracy and stability in large-scale complex networks, and its time complexity in sparse networks is near $O(n)$.

Table 1 lists the different methods of above algorithms to improve COPRA and their time complexity in sparse networks. Obviously, the belonging coefficient is important to the MLPAs. Most algorithms redefine the belonging coefficient, and recently there are still some researchers improving the calculation method of belonging coefficient.

For example, Kouni et al. [19] and Gao et al. [20] respectively proposed new calculation methods of belonging coefficient, and Attal et al. [21] compared the effects of different belonging coefficients in complex networks.

### Table 1. Different methods to improve COPRA and the time complexity in sparse networks of the algorithms.

| Algorithm | Method | Time complexity in sparse networks |
|-----------|--------|-----------------------------------|
| SLPA      | - Label update history | near $O(n)$ |
| BMLPA     | - Rough core algorithm to classify nodes | $O(n \log n)$ |
|           | - Balanced belonging coefficient | |
| CLBLPA    | - Leader Rank algorithm to select core nodes | $O(n \log n)$ |
| LPA-E     | - Redefine belonging coefficient according to information entropy | near $O(n)$ |
|           | - Threshold for filtering labels is $1/|\text{Nb}(x)|$ | |
| DLPA      | - Redefine belonging coefficient according to Jaccard index | near $O(n)$ |
|           | - Threshold for filtering labels is $1/|\text{Nb}(x)|$ | |
|           | - Dominant label | |
| LPANNI    | - Fix node update sequence according to the ascending order of node importance | near $O(n)$ |
|           | - Redefine belonging coefficient according to neighbors’ influence on nodes | |
|           | - Threshold for filtering labels is $1/|L_x(x)|$ | |
|           | - Dominant label | |

However, the threshold for filtering labels is usually not paid enough attention in MLPAs. In the algorithms described above, their thresholds for filtering labels are set to a global parameter (SLPA and BMLPA), $1/|\text{Nb}(x)|$ (LPA-E and DLPA) or $1/|L_x(x)|$ (LPANNI) when node $x$ updates labels. Usually, when a node updates labels, if it receives more different labels, or the belonging coefficients of the received labels are closer, its threshold should be smaller.

Meanwhile, when nodes update labels, nodes in DLPA and LPANNI only receive neighbors’ dominant labels, but nodes in other algorithms receive all labels of neighbors. Compared with the above two methods, it is a better choice for nodes to receive only part of the labels with large belonging coefficients from their neighbors.

In LPANNI, the node update sequence is fixed by the ascending order of node importance. This fixed node update sequence greatly improves the stability of the algorithm, but a better fixed node update sequence can improve stability and accuracy at the same time. Thus, a new fixed node update sequence needs to be designed.

Based on the above analysis, a new multi-label propagation algorithm is proposed in Section III.

### III. THE MULTI-LABEL PROPAGATION ALGORITHM WITH THE DOUBLE-LAYER FILTERING STRATEGY

Compared with LPANNI, MLPA-DF has two main changes. First, MLPA-DF replaces the method of receiving labels and
filtering labels in the LPANNI’s label update step with the double-layer filtering strategy. Second, MLPA-DF uses a different fixed node update sequence from LPANNI.

A. THE DOUBLE-LAYER FILTERING STRATEGY

In the first layer of the double-layer filtering strategy, a threshold \( \theta_1 \) is designed to help nodes filter labels. This layer takes effect after a node receives labels from neighbors and recalculates the label’s belonging coefficients. Only if a label’s belonging coefficient is larger than \( \theta_1 \), the node will set it as a new label. Therefore, if \( \theta_1 \) of a node is smaller, the node tends to select more labels as its new labels. The \( \theta_1 \) of node \( x \) is defined as follows:

\[
\theta_1(x) = g \sqrt{\frac{\sum_{c \in L_r(x)} b^2(c,x)}{|L_r(x)|}}
\]

(1)

where \(|L_r(x)|\) is the number of different labels received from neighbors by \( x \). \( b(c,x) \) is the belonging coefficient between node \( x \) and label \( c \). \( g \) is a manually set parameter.

Because MLPA-DF uses the same method as LPANNI to recalculate the belonging coefficients of received labels by the node, these belonging coefficients are normalized and the value range of \( \sum_{c \in L_r(x)} b^2(c,x) \) is \([1/|L_r(x)|,1]\). The smaller the sum of squares, the closer the belonging coefficients of these labels are, and the smaller \( \theta_1 \) is. \( g \) is a global parameter that affects the \( \theta_1 \) of all nodes. It usually has different values in different networks. The first layer of the double-layer filtering strategy is the same as the label filtering method in LPANNI except for the threshold.

In MLPA-DF, the second layer of the double-layer filtering strategy is to divide nodes into active labels and inactive labels. Only the active labels of neighbors are received by the node that is updating labels. Another threshold \( \theta_2 \) is set to divide the nodes into two categories after a node selects new labels and normalizes new labels’ belonging coefficient. If a label’s belonging coefficient is larger than \( \theta_2 \), the label is an active label. And other labels are inactive labels. The \( \theta_2 \) of node \( x \) is defined as follows:

\[
\theta_2(x) = \frac{1}{|L_n(x)| + 1}
\]

(2)

where \(|L_n(x)|\) is the number of new labels selected by node \( x \).

The average belonging coefficient of the new labels is \( 1/|L_n(x)| \), and labels with belonging coefficients less than the average are usually not important to \( x \). However, if the belonging coefficients of all new labels are different but close, these belonging coefficients are usually close to \( 1/|L_n(x)| \) as well, and all labels may have similar importance to \( x \). If \( \theta_2(x) \) is set to \( 1/|L_n(x)| \), there must be a part of labels whose belonging coefficients are less than the average, then they are classified as inactive labels. For example, a node has three new labels, and their belonging coefficients are 0.34, 0.34, and 0.32 respectively. It is inappropriate to set the label whose belonging coefficient is 0.32 as an inactive label. This problem can be avoided by just making \( \theta_2(x) \) slightly smaller than \( 1/|L_n(x)| \). In fact, the active labels in MLPA-DF play a similar role to the dominant labels in LPANNI.

In order to use the double-layer filtering strategy, the initialization step and termination criterion need to be changed accordingly in MLPA-DF. In initialization step, the initial label of each node is set to the active label. And the termination criterion for updating labels step is only that the number of iterations reaches the upper limit.

B. FIXED NODE UPDATE SEQUENCE

In MLPA, nodes update their labels according to the labels of their neighbors. If a node has a large number of neighbors, it sends its own labels to many nodes, and these labels are usually selected as the new labels by more nodes. In addition, if two neighbors of a node are connected to each other, the node and these two neighbors form a triangle, and the three nodes are likely to select the same labels as their new labels in the same iteration. Therefore, the influence of a node’s labels can be measured by the number of node neighbors and the number of edges between node neighbors. And the greater the influence of the labels of a node, the more likely it is to become the core of the community, so the node centrality (NC) is designed as follows:

\[
NC(x) = |Nb(x)| + \sum_{y \in |Nb(x)|} |Nb(x) \cap Nb(y)|
\]

(3)

where \( Nb(x) \) denotes the neighbor set of node \( x \). \(|Nb(x)|\) denotes the number of neighbors of node \( x \).

NC is used to fix the node update sequence in MLPA-DF. Because labels of a node with a larger NC are propagated to more nodes, new labels of a node usually come from neighbors whose NC is larger than the node. If a node with a small NC update labels first in one iteration, its new labels may be the old labels of a neighbor with a large NC. Unless the new labels of the neighbor remain the same as the old labels, the node with a small NC make an invalid update. Based on this, nodes in MLPA-DF are updated in descending order of NC.

C. THE PROCEDURE OF MLPA-DF AND AN EXAMPLE ON THE SAMPLE NETWORK

Figure 1 is a sample network and MLPA-DF is used to detect overlapping communities on it.

**FIGURE 1.** A sample network.
The complete MLPA-DF is shown in Algorithm 1. The definitions of NNI and belonging coefficient in MLPA-DF are same as those in LPANNI.

![Algorithm 1: MLPA-DF](image)

Figure 4 is the label propagation phase of MLPA-DF on the sample network.

![Figure 4](image)

**TABLE 2. NC of the sample network.**

| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| NC   | 8 | 7 | 4 | 7 | 7 | 4 | 8 | 1 |

**TABLE 3. NNI matrix of the sample network.**

| Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| 1    | -- | 0.48 | -- | 0.48 | 0.48 | -- | 0.47 | -- |
| 2    | 0.53 | -- | 0.53 | 0.54 | -- | -- | -- | -- |
| 3    | -- | 0.67 | -- | 0.67 | -- | -- | -- | -- |
| 4    | 0.53 | 0.54 | 0.53 | -- | -- | -- | -- | -- |
| 5    | 0.54 | -- | -- | -- | 0.54 | 0.54 | -- | -- |
| 6    | -- | -- | -- | 0.69 | -- | 0.68 | -- | -- |
| 7    | 0.42 | -- | -- | -- | 0.43 | 0.42 | -- | 0.42 |
| 8    | -- | -- | -- | -- | -- | -- | 1.0 | -- |

Figure 4 is the label propagation phase of MLPA-DF on the sample network. The label c of the node and its belonging coefficient b are recorded in the form of (c, b), b retains three decimal places. The parameter g is set to 0.8.

Figure 4(a) is the initialization, each node is given a unique label, and the label’s belonging coefficient is set to 1. Figure 4(b) is the result after the first iteration. Labels of each node has changed. In node 5, b of label 5 is 0.330, which is less than th2 of node 5 (the th2 of node 5 is 1/3 at this time), so label 5 is an inactive label and is marked by strikethrough. Figure 4(c) is the result after the fourth iterations. Nodes have been clearly divided into two communities. Node 1 is an overlapping node and joins two communities.
D. TIME COMPLEXITY ANALYSIS

Given a network $G(N,E)$, $n=|N|$, $k$ is the average number of neighbors of each node. $l$ is the average number of labels each node has. $T$ is the maximum number of iterations. The time complexity analysis of MLPA-DF is as follows:

1) Calculating the NC of a node needs to know how many edges there are between the neighbors of the node. The time complexity for calculating NC of a node is $O(k^2)$. The time complexity for calculating NC of all nodes is $O(kn)$.

2) A node needs to consider all labels of all neighbors to update labels once. The time complexity for a node to update labels once is $O(k)$. The time complexity for all nodes to update labels $T$ times is $O(Tkn)$.

The time complexity of other parts in MLPA-DF is the same as the corresponding parts in LPANNI. In sparse networks, $n$ is much larger than $k$, $l$ and $T$, and the time complexity of LPANNI is near $O(n)$. Therefore, the time complexity of MLPA-DF in sparse networks is also near $O(n)$.

IV. EXPERIMENT

A. DATASET

Experiments are conducted on synthetic and real-world networks. LFR benchmark [22] is used to generate synthetic networks. Each LFR benchmark network has correct overlapping community structures. In recent years, more than half of the label propagation algorithms have been experimented on the LFR benchmark networks [13]. Some parameters are needed to generate the LFR benchmark networks. These parameters and their meanings are listed in Table 4. Larger $u$ makes overlapping communities more difficult to detect. $O_u$ and $O_m$ can determine the overlap degree of the network. Two different types of LFR benchmark networks are generated. In the first type of LFR benchmark network, there are many communities in which the number of nodes is less than 10% of all nodes. In the second type of LFR benchmark network, the number of communities is small, but the number of nodes in these communities is not less than 10% and not more than 25% of all nodes. There are 26 different LFR benchmark networks in total. For these networks, $k=10$ and $max_k=50$. Other parameters are listed in Table 5. For different experimental purposes, these networks are divided into four groups. Networks in Group A and B are the first type of network, and Networks in Group C and D are the second type of network. Networks in Group A and C are used to test the performance of the algorithms in two types of networks with the same scale and different parameters. And Networks in Group B and D are used to test the performance of the algorithms in two types of networks with different scales and the same parameters.

Table 6 lists the real-world networks used in the experiment. $|E|$ is the number of edges.

### TABLE 4. The meanings of the parameters.

| Parameter | Meaning |
|-----------|---------|
| $|N|$ | Number of nodes |
| $u$ | Mixing parameter |
| $O_u$ | The number of overlapping nodes |
| $O_m$ | The number of communities joined by each overlapping node |
| $minc$ | Minimum number of nodes in the community |
| $maxc$ | Maximum number of nodes in the community |
| $k$ | Average number of neighbors per node |
| $maxk$ | Maximum number of neighbors of a node |

### TABLE 5. Parameters of different LFR benchmark networks.

| Group | ID | $|N|$ | $u$ | $minc$ | $maxc$ | $O_u$ | $O_m$ |
|-------|----|------|-----|-------|-------|------|------|
| A     | A1 | 1000 | 0.05| 10    | 100   | 2    | 2    |
|       | A2 | 1000 | 0.1 | 10    | 100   | 4    | 4    |
|       | A3 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A4 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A5 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A6 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A7 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A8 | 1000 | 0.15| 10    | 100   | 6    | 6    |
|       | A9 | 1000 | 0.15| 10    | 100   | 6    | 6    |
| B     | B1 | 2000 | 0.05| 10    | 100   | 4    | 4    |
|       | B2 | 4000 | 0.05| 10    | 100   | 4    | 4    |
|       | B3 | 6000 | 0.05| 10    | 100   | 4    | 4    |
|       | B4 | 8000 | 0.05| 10    | 100   | 4    | 4    |
| C     | C1 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C2 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C3 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C4 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C5 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C6 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C7 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C8 | 1000 | 0.05| 10    | 250   | 4    | 4    |
|       | C9 | 1000 | 0.05| 10    | 250   | 4    | 4    |
| D     | D1 | 2000 | 0.05| 10    | 200   | 3    | 3    |
|       | D2 | 4000 | 0.05| 10    | 200   | 3    | 3    |
|       | D3 | 6000 | 0.05| 10    | 200   | 3    | 3    |
|       | D4 | 8000 | 0.05| 10    | 200   | 3    | 3    |

### TABLE 6. Attributes of real-world networks.

| Network   | $|N|$ | $|E|$ |
|-----------|------|------|
| Karate [23] | 34 | 78 |
| Dolphins [24] | 62 | 159 |
| Books [25] | 105 | 441 |
| Football [1] | 115 | 613 |
| Facebook [26] | 4039 | 88,234 |
| Ca-HepPh [27] | 12006 | 118489 |
| Email-Enron [28] | 36692 | 183831 |

COPRA and SLPA are popular MLPAs. LPANNI is the basis of MLPA-DF. Parameters for these algorithms are set as follows. For COPRA, $\nu$ varies from 2 to 8. For SLPA, $r$ varies from 0.05 to 0.5 with an interval of 0.05. For LPANNI, $a$ is set to 3 which is the recommended value in reference [18]. For MLAP-DF, $g$ varies from 0.3 to 1.5. The maximum number of iterations $T$ of these algorithms is 25.

For synthetic networks, the performance of these four algorithms is compared from two aspects: the accuracy of detecting overlapping communities and the stability of the algorithm. NMI [29] and $NMI_{max}$ [30] are used to evaluate the accuracy of different algorithms in detecting communities. NMI and $NMI_{max}$ both reach their best and worst value at 1.
and 0, respectively. Considering the randomness of these algorithms, these experiments are repeated for 50 times under each parameter settings separately, and get the average value and standard deviation of the \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) and \( NMI_{\text{max, std}} \) denote the average of \( NMI \) and \( NMI_{\text{max}} \) respectively. \( NMI_{\text{std and max std}} \) denote the standard deviations of \( NMI \) and \( NMI_{\text{max}} \) respectively. Therefore, for every algorithm, the largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) on each LFR benchmark network are used to evaluate its accuracy in detecting overlapping communities, and the corresponding \( NMI_{\text{std}} \) and \( NMI_{\text{max, std}} \) are used to evaluate the stability. Smaller \( NMI_{\text{std and max std}} \) mean the better stability.

For real-world networks, the accuracy and stability of the four algorithms for detecting overlapping communities are compared. \( Q_v \) [31] is used to evaluate the accuracy of different algorithms in detecting communities due to the lack of correct overlapping community structures of real-world networks. The definition of \( Q_v \) in reference [31] is for directed networks. It needs to be modified to apply to undirected networks. Therefore, the definition of \( Q_v \) in this paper is as follows:

\[
Q_{ov} = \frac{1}{2|E|} \sum_{c \in C} \sum_{p,w \in N} \left[ F(b_{vc}, b_{wc}) A_{vw} - k_p k_w \sum_{i \in N} F(b_{vc}, b_{ic}) \sum_{j \in E} F(b_{vc}, b_{jc}) \right]
\]

(4)

\[
F(b_{vc}, b_{wc}) = \frac{1}{(1 + e^{-f(b_{vc})})(1 + e^{-f(b_{wc})})}
\]

(5)

where \( A_{vw} \) denotes whether nodes \( v \) and \( w \) are connected. If \( v \) and \( w \) are connected, \( A_{vw} \) is 1, otherwise \( A_{vw} \) is 0. \( k_v \) denotes the number of neighbors of node \( v \). \( b_{vc} \) denotes the belonging coefficient of label \( c \) to node \( v \). The function \( f \) is the same as the function recommended in reference [32]:

\[
f(x) = 60x - 30
\]

\[Q_{ov}\] reaches its best and worst value at 1 and 0, respectively. Similar to synthetic networks, these experiments are repeated for 50 times under each parameter settings on each real-world network separately, and get the average and standard deviation of \( Q_{ov} \). \( Q_{ov, av} \) and \( Q_{ov, std} \) are the average and standard deviation of \( Q_{ov} \). The largest \( Q_{ov, av} \) is used to evaluate the accuracy of detecting overlapping communities, and the corresponding \( Q_{ov, std} \) is used to evaluate the stability. Smaller \( Q_{ov, std} \) means the better stability.

All the experiments are conducted on a PC with 1.70 GHz AMD A8-5545M APU and 16 GB RAM. All algorithms are implemented in Python 3.7.

C. EXPERIMENTAL RESULTS ON THE SYNTHETIC NETWORKS

1) COMPARISON OF ACCURACY ON THE LFR BENCHMARK NETWORKS

FIGURES 5, 6, 7 and 8 show the largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) of the algorithms on the LFR benchmark networks in groups A, B, C, and D, respectively. The parameters of these algorithms to get the largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) are listed in Table 7.

TABLE 7. Parameters of different algorithms to get largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) on the LFR benchmark networks.

| ID | COPRA(\(\beta\)) | SLPA(\(\tau\)) | MLPA-DF(\(g\)) |
|----|----------------|----------------|----------------|
| NMI_{\text{av}} | NMI_{\text{max, av}} | NMI_{\text{av}} | NMI_{\text{max, av}} | NMI_{\text{av}} | NMI_{\text{max, av}} |
| A1 | 4  | 3  | 0.1 | 0.05 | 0.8 | 0.7 |
| A2 | 5  | 5  | 0.1 | 0.05 | 0.7 | 0.6 |
| A3 | 8  | 8  | 0.05 | 0.05 | 0.8 | 0.8 |
| A4 | 4  | 3  | 0.05 | 0.05 | 0.8 | 0.8 |
| A5 | 7  | 6  | 0.1 | 0.05 | 0.7 | 0.7 |
| A6 | 7  | 7  | 0.05 | 0.05 | 0.8 | 0.8 |
| A7 | 3  | 3  | 0.25 | 0.05 | 0.7 | 0.8 |
| A8 | 8  | 5  | 0.1 | 0.05 | 0.9 | 0.9 |
| A9 | 8  | 8  | 0.05 | 0.05 | 0.6 | 0.7 |
| B1 | 7  | 6  | 0.05 | 0.05 | 0.7 | 0.7 |
| B2 | 6  | 6  | 0.1 | 0.05 | 0.7 | 0.8 |
| B3 | 6  | 6  | 0.1 | 0.05 | 0.7 | 0.7 |
| B4 | 6  | 6  | 0.1 | 0.05 | 0.7 | 0.7 |
| C1 | 3  | 3  | 0.1 | 0.1 | 0.6 | 0.6 |
| C2 | 6  | 6  | 0.15 | 0.05 | 0.6 | 0.6 |
| C3 | 6  | 6  | 0.15 | 0.1 | 0.6 | 0.6 |
| C4 | 3  | 3  | 0.2 | 0.2 | 0.9 | 0.7 |
| C5 | 4  | 4  | 0.1 | 0.1 | 0.6 | 0.7 |
| C6 | 6  | 6  | 0.15 | 0.05 | 0.7 | 0.7 |
| C7 | 3  | 3  | 0.2 | 0.1 | 0.7 | 0.7 |
| C8 | 3  | 3  | 0.15 | 0.1 | 0.8 | 0.8 |
| C9 | 3  | 3  | 0.2 | 0.1 | 0.6 | 0.6 |
| D1 | 3  | 3  | 0.2 | 0.1 | 0.6 | 0.6 |
| D2 | 3  | 3  | 0.2 | 0.15 | 0.5 | 0.5 |
| D3 | 2  | 2  | 0.2 | 0.15 | 0.4 | 0.4 |
| D4 | 2  | 2  | 0.25 | 0.15 | 0.5 | 0.5 |

FIGURE 5. The largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) of the algorithms on the LFR benchmark networks in Group A.

FIGURE 6. The largest \( NMI_{\text{av}} \) and \( NMI_{\text{max, av}} \) of the algorithms on the LFR benchmark networks in Group B.
In Figure 5, LPANNI, SLPA and COPRA have similar performance in terms of largest $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ on the LFR benchmark networks in Group A. When $O_m=6$, SLPA performs slightly better than the other two algorithms. When $O_m=4$, the performance of LPANNI is slightly better than the other two algorithms. Meanwhile, the performance of MLPA-DF is obviously better than the other three algorithms. This shows that MLPA-DF has better accuracy in detecting overlapping communities than the other three algorithms on the first type of LFR benchmark networks that have a large number of communities with few nodes.

In Figure 6, although the scale of the experimental network is constantly increasing, the largest $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ of the four algorithms have not changed much. This shows that on the first type of LFR benchmark network, the scale of the network has little effect on the accuracy.

In Figure 7, MLPA-DF performs best except on the networks with $O_m=2$, and LPANNI performs worst on the most networks. This means that MLPA-DF can also have good accuracy on the second type of LFR benchmark networks where communities have a large number of nodes, while LPANNI is not suitable for this type of LFR benchmark networks.

In Figure 8, the change of the network scale has some influence on the $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ of the algorithms, but MLPA-DF still has the best performance in most cases.

Combining Figure 5 and Figure 7, in most cases, the $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ of the algorithms decrease with the increase of $O_m$. And the $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ of MLPA-DF obviously decrease more slowly than other algorithms. This shows that for LFR benchmark networks where overlapping nodes tend to join many communities, MLPA-DF has a greater advantage than other algorithms.

In general, MLPA-DF can detect overlapping communities more accurately than other algorithms on most LFR benchmark networks, regardless of the type of networks.

2) COMPARISON OF STABILITY ON THE LFR BENCHMARK NETWORKS

$\text{NMI}_{\text{std}}$ and $\text{NMI}_{\text{max-std}}$ are used to analyze the stability of the algorithm. Figure 9 is the distribution of $\text{NMI}_{\text{std}}$ and $\text{NMI}_{\text{max-std}}$ when these algorithms get the largest $\text{NMI}_{\text{avg}}$ and $\text{NMI}_{\text{max-avg}}$ on 26 LFR benchmark networks in Table 5.

On most LFR benchmark networks, $\text{NMI}_{\text{std}}$ and $\text{NMI}_{\text{max-std}}$ of SLPA and COPRA are distributed between 0 and 0.05. All $\text{NMI}_{\text{max-std}}$ of LPANNI and MLPA-DF are close to 0. Therefore, for LFR benchmark networks, COPRA and SLPA has poor stability, LPANNI and MLPA-DF have good stability, and the stability of SLPA is better than COPRA, worse than LPANNI and MLPA-DF. This also means the new fixed node update sequence and the double-layer filtering strategy do not reduce the stability of the algorithm.

3) EXPERIMENTAL RESULTS ON THE SAMPLE LFR BENCHMARK NETWORK

The following shows the results of four algorithms for detecting overlapping communities on a sample LFR benchmark network.

The sample LFR benchmark network has 90 nodes, 3 communities and 9 overlapping nodes. Each overlapping node join two communities. There are about 1/3 of all nodes in each community. The correct community structure of the sample LFR benchmark network is shown in Figure 10. Figure 11 is the detection results of four algorithms.
In Figure 10 and Figure 11, the color of each node indicates the community it joins and different colored areas indicate different communities. If a node joins multiple communities, it has multiple colors.

In Figure 11, MLPA-DF and SLPA detect 3 communities, but COPRA and LPANNI detect 4 and 2 communities respectively. The detection result of LPANNI differs the most from the correct overlapping community structure. This shows that LPANNI is not suitable for this type of network, which is consistent with the above conclusion. Meanwhile, the detection of overlapping communities usually has the following three problems: One community is mistakenly divided into more than one community, some communities mistakenly become one community, and nodes join the wrong community. On this network, MLPA-DF avoids the first two problems and is better than other algorithms in dealing with the last problem. Compared with other algorithms, MLPA-DF gets the correct number of communities and actively looks for overlapping nodes, although there are some errors.

D. Experimental results on the real-world networks

Table 8 lists the performance of the algorithms on the real-world networks. COPRA and SLPA have large $Q_{w, avg}$ in all real-world networks, so the stability of these two algorithms is poor on these real-world networks. The $Q_{w, std}$ of LPANNI and MLPA-DF are close to 0, so they have good stability on the real-world networks.

Except for Karate networks, MLPA-DF has the largest $Q_{w, avg}$. On Karate network, the $Q_{w, avg}$ of MLPA-DF is not much different from the largest $Q_{w, avg}$. This shows that MLPA-DF has good accuracy on the real-world networks.

In addition, the overlapping community structure of real-world networks can be further analyzed based on the detection results of the largest $Q_{w, avg}$ in Table 8. Consistent with the experiments on the synthetic network, the communities in the real-world network detection results can be divided into large-communities (the number of nodes is not less than 10% of all nodes) and small-communities (the number of nodes is less than 10% of all nodes). Then the $N_L$ and $N_s$ in the detection result of each real-world network are listed in Table 9. $N_L$ denotes the proportion of nodes joining the large-communities to all nodes and $N_s$ denotes the proportion of nodes joining the small-communities to all nodes.

In small-scale networks, there are almost all large-communities. This is probably because the number of nodes is small and there is no need to split out too many communities. In large-scale networks, both types of communities exist, and if the scale is large enough, the proportion of nodes in small-communities may be higher than the proportion of nodes in large-communities. This also explains why LPANNI performs better in large-scale networks.

**TABLE 8. Performance of different algorithms on the real-world networks.**

| Network | Karate | Dolphins | Books | Football | Facebook | Ca-HepPh | Email-Enron |
|---------|--------|----------|-------|----------|----------|----------|-------------|
| COPRA   | $Q_{w, avg}$ | 0.428 | 0.744 | 0.842 | 0.696 | 0.938 | 0.781 | 0.713 |
|         | $Q_{w, std}$ | 0.170 | 0.011 | 0.011 | 0.021 | 0.005 | 0.005 | 0.022 |
| SLPA    | $Q_{w, avg}$ | 0.760 | 0.775 | 0.833 | 0.698 | 0.926 | 0.781 | 0.719 |
|         | $Q_{w, std}$ | 0.010 | 0.017 | 0.023 | 0.016 | 0.010 | 0.005 | 0.018 |
|         | $r$ | 0.45 | 0.45 | 0.4 | 0.25 | 0.20 | 0.50 | 0.45 |
| LPANNI  | $Q_{w, avg}$ | 0.756 | 0.764 | 0.749 | 0.694 | 0.915 | 0.804 | 0.768 |
|         | $Q_{w, std}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |
|         | $Q_{v, std}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |
|         | $Q_{g, std}$ | 0.756 | 0.785 | 0.851 | 0.732 | 0.970 | 0.807 | 0.777 |
| MLPA-DF | $Q_{w, avg}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |
|         | $Q_{w, std}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |
|         | $Q_{v, std}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |
|         | $Q_{g, std}$ | 0 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 | <0.001 |

**TABLE 9. $N_L$ and $N_s$ of different real-world networks.**

| Network     | $N_L$ | $N_s$ | Algorithm       |
|-------------|-------|-------|-----------------|
| Karate      | 100%  | 0%    | SLPA ($r=0.45$) |
| Dolphins    | 96.80%| 3.20% | MLPA-DF ($g=0.9$) |
| Books       | 100%  | 0%    | MLPA-DF ($g=1.2$) |
| Football    | 73.90%| 26.10%| MLPA-DF ($g=1.2$) |
| Facebook    | 74.40%| 25.70%| MLPA-DF ($g=1.2$) |
| Ca-HepPh    | 36.70%| 64.20%| MLPA-DF ($g=1.4$) |
| Email-Enron | 43.40%| 56.70%| MLPA-DF ($g=1.5$) |
networks. MLPA-DF does not have this limitation, so it solves the problem of insufficient accuracy of LPANNI on small-scale real-world networks.

Considering both accuracy and stability on these real-world networks, COPRA has poor accuracy and stability, SLPA has good accuracy but poor stability, LPANNI has good stability, and MLPA-DF can achieve good accuracy and stability at the same time. Therefore, MLPA-DF can perform well on real-world networks.

V. CONCLUSION
In this paper, a new multi-label propagation algorithm, MLPA-DF, is proposed to detect overlapping communities in complex networks. MLPA-DF is an improved version of LPANNI. In MLPA-DF, the double-layer filtering strategy and node centrality are used to help nodes filter labels and fix node update sequence respectively.

The theory analysis show that MLPA-DF has the near linear time complexity. MLPA-DF and three popular MLPPs (COPRA, SLPA and LPANNI) are tested on both synthetic networks and real-world networks. Experimental results show that MLPA-DF can improve the accuracy for detecting overlapping communities without reducing stability. And MLPA-DF solves the problem of insufficient accuracy of LPANNI in some networks.

However, MLPA-DF still has two problems: one is that it needs to set different parameters \( g \) on different networks based on experience. The other is that it cannot detect overlapping communities on directed or weighted networks. Solving these two problems will be our important work in the future.

REFERENCES
[1] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, Jun. 2002, doi: 10.1073/PNAS.122653799.
[2] J. Xie, S. Kelley and B.K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and competitive study,” ACM Comput. Surv., vol. 45, no. 4, p. 43, Aug. 2013, doi: 10.1145/2468543.2506167.
[3] G. Palla, I. Derényi, I. Farkas and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” Nature, vol. 435, no. 7043, pp. 814–818, Jun. 2005, doi: 10.1038/NATURE03607.
[4] I. J. Farkas, D. Dabel, G. Palla and T. Vicsek, “Weighted network modules,” New J. Phys., vol. 9, no. 6, pp. 180–180, Jun. 2007, doi: 10.1088/1367-2630/9/6/180.
[5] Y. Y. Ahn, J. P. Bagrow and S. Lehmann, “Link communities reveal multiscale complexity in networks,” Nature, vol. 466 no. 7307, pp. 761–764, Aug. 2010, doi: 10.1038/NATURE09182.
[6] T. S. Evans and R. Lambiotte, “Line Graphs of Weighted Networks for Overlapping Communities,” Eur. Phys. J. B, vol. 77, no. 2, pp. 265–272, Sep. 2010, doi: 10.1140/EPJB/E2010-00261-8.
[7] A. Lanchinichetti, F. Radicchi, J. J. Ramasco and S. Fortunato, “Finding statistically significant communities in networks,” Phys. One, vol. 6, no. 4, pp. 1–18, Apr. 2011, doi: 10.14711/JOURNAL_PONE.0018961.
[8] R. Shang, W. Zhang, J. Zhang, J. Feng and L. Jiao, “Local community detection based on higher-order structure and edge information,” Physica A, vol. 587, p. 126513, Feb. 2022, doi: 10.1016/j.physa.2021.126513.
[9] S. Zhang, R. Wang and X. Zhang, “Uncovering Fuzzy Community Structure in Complex Networks,” Phys. Rev. E, vol. 76, no. 4, pp. 46103–46103, Oct. 2007, doi: 10.1103/PHYSREве.76.046103.
[10] S. Gregory, “Finding overlapping communities in networks by label propagation,” New J. Phys., vol. 12, no. 10, p. 103018, Oct. 2010, doi: 10.1088/1367-2630/12/10/103018.
[11] U. N. Raghavan, R. Albert and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Phys. Rev. E, vol. 76, no. 3, pp. 36106–36106, Sep. 2007, doi: 10.1103/PHYSREве.76.036106.
[12] Z. Wu, Y. Lin, S. Gregory, H. Wan and S. Tian, “Balanced Multi-Label Propagation for Overlapping Community Detection in Social Networks,” J. Comput. Sci. Technol., vol. 27, no. 3, pp. 468–479, May 2012, doi: 10.1007/s11390-012-1236-X.
[13] S. E. Garza and S. E. Schaeffer, “Community Detection with the Label Propagation Algorithm: A Survey,” Physica A, vol. 534, p. 122058, Nov. 2019, doi: 10.1016/j.physa.2019.122058.
[14] J. Xie, B. K. Szymanski and X. Liu, “Sipa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process,” in Proc. IEEE Int. Conf. Data Min., Vancouver, BC, Canada, 2011, pp. 344–349, doi: 10.1109/ICDMW.2011.184.
[15] H. Sun, J. Huang, Y. Tian, Q. Song and H. Liu, “Detecting Overlapping Communities in Networks via Dominant Label Propagation,” Chin. Phys. B, vol. 24, no. 1, p. 18703, Jan. 2015, doi: 10.1088/1674-1056/24/1/018703.
[16] N. Chen, Y. Liu, H. Chen and J. Cheng, “Detecting communities in social networks using label propagation with information entropy,” Physica A, vol. 471, pp. 788–798, Apr. 2017, doi: 10.1016/j.physa.2016.12.047.
[17] S. Liu, F. Zhu, H. Liu and Z. Du, “A core leader based label propagation algorithm for community detection,” China Communications, vol. 13, no. 12, pp. 97–106, Dec. 2016, doi: 10.1109/CC.2016.7897535.
[18] 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, Sep. 2019, pp. 1736–1749, doi: 10.1109/TKDE.2018.2866424.
[19] J. B. Kouni, W. Karoui and L. B. Rondhane, “Node Importance based Label Propagation Algorithm for overlapping community detection in networks,” Expert Syst. Appl., vol. 162, p. 113020, Dec. 2020, doi: 10.1016/J.ESEA.2019.113020.
[20] R. Gao, S. Li, X. Shi, Y. Liang and D. Xu, “Overlapping Community Detection Based on Membership Degree Propagation,” Entropy, vol. 23, no. 1, p. 15, Jan. 2021, doi: 10.3390/e23010015.
[21] J.-P. Aitai, M. Maleki and M. Zolghadri, “Overlapping community detection using core label propagation algorithm and belonging functions,” Appl. Intell., vol. 51, no. 11, pp. 8067–8087, Nov. 2021, doi: 10.1007/S10489-021-02250-4.
[22] A. Lanchinichetti, S. Fortunato and F. Raddichi, “Benchmark graphs for testing community detection algorithms,” Phys. Rev. E, vol. 78, no. 4, pp. 46110, Oct. 2008, doi: 10.1103/PHYSREве.78.046110.
[23] W. W. Zachary, “An information flow model for conflict and fission in small groups,” J. Anthropol. Res., vol. 33, no. 4, pp. 452–473, 1977, doi: 10.1063/1.3629752.
[24] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten and S. M. Dawson, “The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations,” Behav. Ecol. Sociobiol., vol. 54, no. 4, pp. 396–405, Sep. 2003, doi: 10.1007/s00265-003-0661-Y.
[25] W. Krebs, 2004, “A network of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com,” distributed by M. E. J. Newman, http://www-personal.umich.edu/~mejn/netdata/.
[26] J. Leskovec and J. J. McAuley, “Learning to Discover Social Circles in Ego Networks,” presented at 26th Annul. Conf. Neural Inf. Process. Syst., Dec. 2012, [Online]. Available: http://snap.stanford.edu/data/ego-links-Facebook.html.
[27] J. Leskovec, J. Kleinberg and C. Faloutsos, “Graph evolution: Densification and shrinking diameters,” ACM Trans. Knowl. Discov. Data, vol. 1 no. 1, p. 2, Mar. 2007, doi: 10.1145/1217299.1217301.
[28] 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,” 2008, [Online].
Available: [http://snap.stanford.edu/data/email-Enron.html](http://snap.stanford.edu/data/email-Enron.html)

[29] A. Lancichinetti, S. Fortunato and J. Kertesz, “Detecting the overlapping and hierarchical community structure in complex networks,” *New J. Phys.*, vol. 11, no. 3, p. 33015, Mar. 2009, doi: 10.1088/1367-2630/11/3/33015

[30] A. F. McDaid, D. Greene and N. J. Hurley, “Normalized Mutual Information to Evaluate Overlapping Community Finding Algorithms,” 2011. [Online]. Available: [https://arxiv.org/abs/1110.5813v3](https://arxiv.org/abs/1110.5813v3)

[31] V. Nicosia, G. Mangioni, V. Carchiolo and M. Malgeri, “Extending the definition of modularity to directed graphs with overlapping communities,” *J. Stat. Mech.*, vol. 2009, no. 3, p. 3024, Mar. 2009, doi: 10.1088/1742-5468/2009/03/P03024

[32] V. Nicosia, G. Mangioni, V. Carchiolo and M. Malgeri, “Extending Modularity Definition for Directed Graphs with Overlapping Communities,” 2008. [Online]. Available: [https://arxiv.org/abs/0801.1647](https://arxiv.org/abs/0801.1647)