Efficient Overlapping Community Detection in Huge Real-world Networks

Zhihao Wu*,[a], Youfang Lin[a], Huaiyu Wan[a], Shengfeng Tian[a], Keyun Hu[b]

[a]*School of Information and Technology, Beijing Jiaotong University, Beijing 100044, China*
[b]*China Mobile Research Institute, Beijing, China*

Abstract

The detection of overlapping community structure in networks can give insight into the structures and functions of many complex systems. In this paper, we propose a simple but efficient overlapping community detection method for very large real-world networks. Taking a high-quality, non-overlapping partition generated by existing, efficient, non-overlapping community detection methods as input, our method identifies overlapping nodes between each pair of connected non-overlapping communities in turn. Through our analysis on modularity, we deduce that, to become an overlapping node without demolishing modularity, nodes should satisfy a specific condition presented in this paper. The proposed algorithm outputs high quality overlapping communities by efficiently identifying overlapping nodes that satisfy the above condition. Experiments on synthetic and real-world networks show that in most cases our method is better than other algorithms either in the quality of results or the computational performance. In some cases, our method is the only one that can produce overlapping communities in the very large real-world networks used in the experiments.

Key words: Complex Networks, Overlapping Community Detection, Overlapping Nodes

1. Introduction

As a mathematical tool, a network can represent many complex systems effectively. For example, a social network represents relationships among people, biological networks represent interactions of molecules or proteins and the WWW is formed of web pages and hyperlinks. These networks have some common features, such as power law degree distribution, clustering and community structures. Communities indicate groups of nodes such that nodes within a group are much more connected to each other than to the rest of the network. Although there is no accurate definition of a community, it exists in various systems, such as organizations in social networks, protein complexes in biological networks or a group of web pages with similar topics on the WWW.

Detecting communities is very important to understand the structure, function and evolution of various systems [1]. To solve this problem, many methods have come forth in recent years, such as betweenness-based methods, similarity-based methods, modularity-based methods, and some other methods based on information theory and random walk. One can refer to Ref. [2] for a detailed review about these methods. Most of the above methods only detect non-overlapping communities, i.e., a node can only belong to one community; however, in some cases, a node may belong to multiple communities. For example, a researcher may belong to more than one research group, or a protein may exist in multiple complexes.

*Corresponding author at: School of Information and Techology, Beijing Jiaotong University, Beijing 100044, China. Tel.:+86 10 51688648; fax: +86 10 51840526.

Email address: zhihaowu@bjtu.edu.cn (Zhihao Wu), yflin@bjtu.edu.cn (Youfang Lin), huaiyuwan@bjtu.edu.cn (Huaiyu Wan), sftian@bjtu.edu.cn (Shengfeng Tian) and hukeyun@chinamobile.com (Keyun Hu)

Taking this situation into account, many algorithms output communities that can overlap [3, 4, 5, 6, 7, 8]. CFinder [3] detects communities through k-clique percolation. Because a node may belong to multiple k-cliques, this method guarantees communities can overlap. LFM [4] is a local algorithm based on local optimization of a fitness function to find overlapping communities that are revealed by peaks in the fitness histogram. COPRA [5] employs a label propagation technology to find overlapping communities, and in its results each node can belong to v communities at the most. Here, v is a tunable parameter. GCE [6] is a two steps algorithm. In the first step, the algorithm identifies distinct cliques as seeds and expands these seeds by greedily optimizing a local fitness function in the second step. OSLOM [7] is a method based on the local optimization of a fitness function expression the statistical significance of clustering with respect to random fluctuations. Another kind of methods to find overlapping communities is to cluster links, such as Ref. [8]. The first step of this method is to construct a line graph of the original network. Then a non-overlapping community detection algorithm can be used to find link communities. Since vertices can belong to multiple links, it guarantees communities can overlap with each other.

Thanks to developments in computing and communications technology, the typical size of large-scale networks, such as user networks of Facebook, mobile phone networks or web networks now count in millions or even billions of nodes.The above overlapping community detection methods have advantages in different aspects, but until now, we find that all these methods suffer a common problem of high computational complexity. Although some methods can process large sparse synthetic networks, few current algorithms can produce overlapping communities on huge real-world networks. Some features of real-world networks might not be contained in synthetic networks. As a result, such large-scale real-world data sets demand new, efficient methods.

From earlier methods, such as the GN algorithm [9], which can only process small networks, to current methods, such as the Infomap [10], BGLL [11], RAK [12] and RG [13], which successfully find communities in very large networks, non-overlapping community detection methods have reached a high level despite the quality or computational performance [14]. After careful inspection of modularity, we deduce that for overlapping community detection, high-quality, non-overlapping community structures almost already contain basic community structures, and we only need to consider some nodes that satisfy a certain condition. Based on this deduction, we plan to propose a fast overlapping community detection method for huge real-world networks.

The rest of this paper is organized as follows. Section 2 explains the design of our new overlapping community detection algorithm. Experimental results of synthetic and real-world networks are shown in section 3. Conclusions appear in section 4.

2. Method

2.1. Definitions

In this paper, we just consider single-edge networks, in which all links must have two different end points. Given an unweighted and undirected graph $G(V, E)$, V represents the node set, E represents the edge set and C_i represents the node set of community i.

Definition 1. For a given node $v \in V$, $N(v) = \{u|(u, v) \in E\}$, we call $N(v)$ the set of all neighbors of v.

Definition 2. For a given node $v \in C_i$, $N_{ii}(v) = N(v) \cap C_i$, we call $N_{ii}(v)$ the set of all neighbors of v in community i, and $N_{ii}^{no}(v)$ the set of all non-overlapping neighbors of v in community i.

Definition 3. For a given node $v \in C_i$ and a community j, $N_{ij}(v) = N(v) \cap C_j$, we call $N_{ij}(v)$ the set of all neighbors of v in community j, and $N_{ij}^{no}(v)$ the set of all non-overlapping neighbors of v in community j.

Definition 4. For two given communities C_i and C_j, $B_{ij} = \{v|v \in C_i, \exists u \in C_j \text{ and } (v, u) \in E\}$ is the boundary node set of community i connecting to community j.

Definition 5. Let $CG(CV, CE)$ be the community graph. CV is the set of communities and $CE = \{(C_i, C_j)|\exists(u, v) \in E, u \in C_i, v \in C_j\}$, is the edge set of CG.

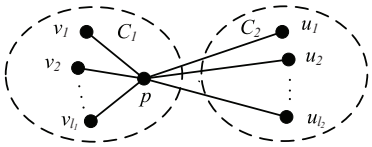

Figure 1: A diagram of boundary node p between two non-overlapping communities.

2.2. The Condition for Overlapping Nodes

Until now, no one has given a comprehensive definition of overlapping nodes. A typical way to define overlapping nodes is to declare that in the output of overlapping community detection algorithms, nodes with multiple community labels are overlapping. In this paper, our rule to judge overlapping nodes is that, when nodes in the generated partition become overlapping nodes, the quality of the community structure does not decrease.

To assess the quality of the community structures, a quality function that evaluates the relative density of edges within communities and between communities, such as Newman's modularity Q [15] and Shen's overlap modularity EQ [16], is usually used.

$$Q = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - (\frac{d_c}{2m})^2 \right]. \tag{1}$$

Here, n_c is the number of communities, l_c is the number of edges joining vertices of community c, d_c is the sum of the degrees of nodes of c and m is the number of edges of the networks.

$$EQ = \frac{1}{2m} \sum_i \sum_{v \in C_i, w \in C_i} \frac{1}{O_v O_w} \left[A_{vw} - \frac{k_v k_w}{2m} \right], \tag{2}$$

where O_v and O_w represent the number of communities to which node v and w belongs, and k_v and k_w are the degree of node v and w, respectively.

In most cases, high-quality, non-overlapping partitions already contain the basic community structures of networks. For the problem of overlapping community detection, we can find overlapping nodes on the basis of a generated partition to form overlapping communities.

Based on the theory of modularity, for a high-quality, non-overlapping partition, moving a node from its community to any other community should not cause obverse increment in the modularity. From this point, we will discuss the condition for overlapping nodes as follows.

First, we will analyse a simple case. As shown in Fig. 1, p is a boundary node between disjoint community C_1 and C_2. The total node degree of community C_1 and C_2 are d_{c_1} and d_{c_2}, respectively. We let $l_1 = N_{11}(p)$ and $l_2 = N_{12}(p)$. If we move node p from community C_1 to C_2, the change of modularity for C_1 is:

$$dQ_1 = -\frac{l_1}{m} + \frac{k_p(2d_{c_1} - k_p)}{(2m)^2}. \tag{3}$$

The change of modularity for C_2 is:

$$dQ_2 = \frac{l_2}{m} - \frac{k_p(2d_{c_2} + k_p)}{(2m)^2}. \tag{4}$$

If node p becomes an overlapping node between C_1 and C_2, according to the definition of EQ, we can get

$$dEQ = \frac{1}{2}(dQ_1 + dQ_2) = \frac{l_2 - l_1}{2m} + \frac{k_p(d_{c_1} - d_{c_2} - k_p)}{(2m)^2}. \tag{5}$$

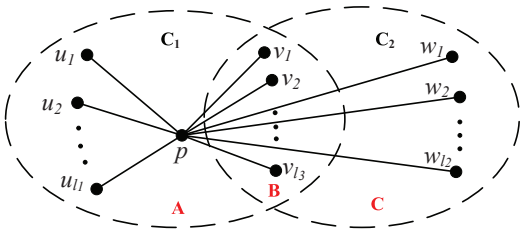

Figure 2: A general diagram of node p with possible overlapping neighbors.

For large-scale networks, $\frac{k_p(d_{c_1}-d_{c_2}-k_p)}{2m^2}$ must be very close to 0. So, we can get Eq.6.

$$dEQ \approx \frac{l_2 - l_1}{2m}. \tag{6}$$

Thus, if a boundary node p between two disjoint communities has a balanced number of connections to each community and we take it as an overlapping node, the modularity of the community structure will not decrease or will only decrease slightly.

Next, we will give a more general case. As shown in Fig. 2, there are two overlapping communities and three node sets: A, B and C. $B = C_1 \cap C_2$, $A = C_1 - B$ and $C = C_2 - B$. Node p, which belongs to community C_1, connects l_1 non-overlapping neighbors in its current community, l_2 non-overlapping neighbors in community C_2 and l_3 overlapping neighbors between the two communities. If l_3 is equal to 0, it becomes the simple case in Fig.1. If l_2 is equal to 0, node p is an inner node. Otherwise, p is a boundary node. In the general case, if node p becomes an overlapping node between community C_1 and C_2, the modularity will change in both communities. For community C_1, the change of overlap modularity is

$$dEQ_1 = -\frac{1}{2}\left[\frac{l_1}{m} - \frac{k_p(2d_A - k_p)}{(2m)^2}\right] - \frac{1}{4}\left[\frac{l_3}{m} - \frac{2k_p d_B}{(2m)^2}\right]. \tag{7}$$

For community C_2, the change of overlap modularity is

$$dEQ_2 = \frac{1}{2}\left[\frac{l_2}{m} - \frac{k_p(2d_C + k_p)}{(2m)^2}\right] + \frac{1}{4}\left[\frac{l_3}{m} - \frac{2k_p d_B}{(2m)^2}\right]. \tag{8}$$

The sum of the two part is

$$dEQ = dEQ_1 + dEQ_2 = \frac{l_2 - l_1}{2m} + \frac{k_p(d_A - k_p - d_C)}{(2m)^2}. \tag{9}$$

where d_A, d_B and d_C are the total degrees of all nodes in each node set.

According to the above assumption, still we can get

$$dEQ \approx \frac{l_2 - l_1}{2m}. \tag{10}$$

Thus, we obtain the *Condition for Overlapping Nodes (CON)* as shown in Eq.10. This deduced condition provides the main evidence for our following design of overlapping community detection algorithm.

2.3. CON-based overlapping community detection algorithm

Based on the deduced *Condition for Overlapping Nodes*, we propose an efficient, overlapping community detection algorithm, named CONA[1] as shown in Algorithm 2. CONA has two steps for each pair of

[1]The implementation of CONA is avilable for download on the webpage http://dev.bjtu.edu.cn/cona/

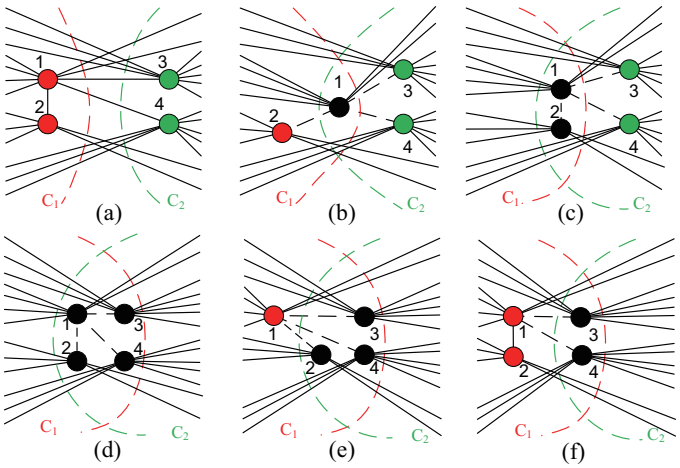

Figure 3: An example of the two procedures of adding and deleting overlapping nodes. $\varphi = 0.55$

communities. In the first step it identifies overlapping nodes from the boundary node set. After that inner overlapping nodes will be detected.

For boundary nodes, from the above equation, we can see if l_2 is larger than l_1, they can be overlapping nodes. However, since one can only pursue a local maximum modularity, the condition of Eq.10 may be too strict. Therefore, we give a relaxed index shown in Eq.11. If $p_{ij}(v)$ is larger than the threshold parameter φ, v is recognized as an overlapping node between C_i and C_j. Thus, if node v connects comparatively balanced number of non-overlapping neighbors in the two communities, even though l_2 may be smaller than l_1, the modularity of community structure will not decrease or will only decrease slightly.

$$p_{ij}(v) = \frac{|N_{ij}^{no}(v)|}{|N_{ii}^{no}(v)|}. \tag{11}$$

For each pair of communities, to find boundary overlapping nodes, CONA will excute the following two procedures repeatedly as shown in Algorithm 1. The first procedure is to find overlapping nodes one by one, until there is no node satisfying the condition for overlapping node. Each time we find the node v with maximum p_{ij}. If $p_{ij}(v)$ is not less than φ, v will be added to overlapping node set. The second procedure is to check whether some found overlapping nodes have become non-overlapping nodes for the detection of new overlapping nodes. In this procedure, we will find and delete the node v with minimum p_{ij} in the current overlapping node set again and again, until $p_{ij}(v)$ is not less than the threshold parameter.

Fig. 3 is an example of the above procedures with $\varphi = 0.55$. Fig.3 (a) shows two disjoint communities and four nodes that may become overlapping nodes. Here we assume that other boundary nodes except these four nodes only have very small p_{ij}. In Fig.3 (a), there are three nodes with maximum p_{ij}: node 1, node 3 and node 4. Suppose we randomly choose node 1 as the first one to be overlapping node. Thus we get Fig.3 (b). At this time, the node with maximum p_{ij} becomes node 2. Node 2 has two non-overlapping neighbors in the other community and three non-overlapping neighbors in its own community. So $p_{12}(2) = 0.67$ and it is larger than $p_{21}(3)$ and $p_{21}(4)$, which are both equal to 0.6. From Fig.3 (c) we can see both node 3 and node 4 can be overlapping nodes. So we get Fig.3 (d). At this time, no more nodes can be overlapping nodes.

Then the second procedure starts. $p_{12}(1)$, which is equal to 0.5, is the minimum p_{ij} in all current overlapping nodes between community C_1 and C_2 and is less than φ. As shown in Fig.3 (e) node 1 is deleted from overlapping node set. Meanwhile, since node 1 becomes non-overlapping node, the number of non-overlapping neighbors of node 2 in its own community becomes four, and $p_{12}(2)$ becomes 0.5. So node 2 also becomes non-overlapping node. Since node 1 and 2 become non-overlapping nodes, $p_{21}(3)$ and $p_{21}(4)$ become larger, and they won't become non-overlapping nodes.

When repeating the above two procedures, no membership changes. Thus node 3 and node 4 are the final boundary overlapping nodes between the two communities.

After the above procedure stops, CONA begins to find overlapping nodes from the inner node set of the two communities. This phase is very simple. Every inner node without non-overlapping neighbors will be insert into overlapping node set.

Algorithm 1: Identifying Overlapping boundary nodes

Input: network $G(V, E)$, partition P, community pair C_i and C_j, threshold parameter φ
Output: overlapping boundary node set

```
set_B = B_ij ∪ B_ji;
flag_1 = true;
while flag_1 do
    flag_1 = false;
    sort p_ij of all nodes of current boundary node set set_B;
    if p_ij^max ≥ φ then
        insert the node with maximum p_ij to set_O and delete it from set_B;
        update related variables;
        flag_1 = true;
        Go to step 5;
    sort p_ij of all nodes of current overlapping node set set_O;
    if p_ij^min < φ then
        delete the node with minimum p_ij from set_O and insert it to set_B;
        update related variables;
        flag_1 = true;
        Go to step 11;
put all inner nodes of the two communities in set_I;
foreach node v in set_I do
    if (v ∈ C_i and N_ii^no(v) = 0) or (v ∈ C_j and N_jj^no(v) = 0) then
        insert v to set_O;
return set_O;
```

Algorithm 2: CONA algorithm

Input: network $G(V, E)$, partition P, threshold parameter φ
Output: overlapping communities

```
compute in and out community degree d_ii and d_ij for all nodes, boundary node sets B for each pair of
communities and community graph CG;
foreach edge (C_i, C_j) in CE do
    call Algorithm 1;
delete the communities that are totally contained by others;
```

To process more efficiently, CONA computes the initial varibles in one-pass scan, including N_{ii}, N_{ij} and the boundary node sets B for each pair of communities. At the same time, a community graph CG is also constructed. After that, one can call Algorithm 1 to discovery overlapping nodes for each pair of connected communities. After identifying overlapping nodes between each pair of communities, there may be some communities that are contained by others. So the final step of CONA is to delete those totally contained communities.

2.4. Convergence Analysis

Here, we will analysis the convergence of the phase of detecting boundary overlapping nodes. In this phase of CONA, two procedures are excuted repeatedly. In the first procedure, non-overlapping nodes may

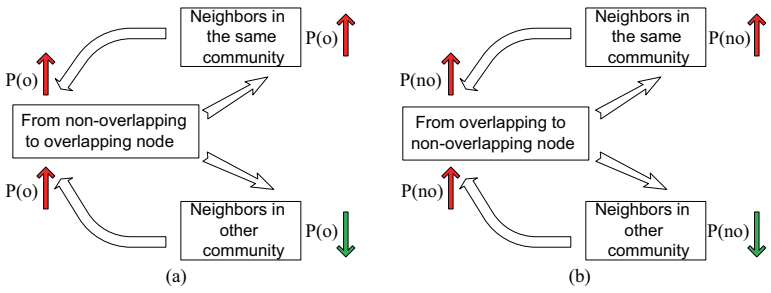

Figure 4: A diagram of convergence analysis. $P(o)$ stands for the possibility of being overlapping node and $P(no)$ represents the possibility of being non-overlapping node.

become overlapping nodes and in the second one, overlapping nodes become non-overlapping nodes. We will analyse the two cases respectively.

In the first case as shown in Fig.4 (a), when a non-overlapping node v becomes overlapping one, it will impact two kinds of nodes. For nodes in the same community, $P(o)$, which stands for the possibility of being overlapping node, will become larger, because l_1 in Eq.10 will be smaller than before. While for nodes in the other community, $P(o)$ will become smaller, because l_2 in Eq.10 will be smaller than before. In a further step, the impact to other nodes will be back to node v. The impact from nodes in the same community are still positive in $P(o)$. Since two negatives make a positive, the impacts from nodes in the other community are also positive in $P(o)$.

The situation in Fig.4 (b) is the same with that in Fig.4 (a). When an overlapping node v becomes non-overlapping one, through the two paths shown in the figure, node v will also get positive impact in $P(no)$ for both cases. Here $P(no)$ represents the possibility of being non-overlapping node.

Based on the above analysis, we can conclude that there is no oscillation in the procedure of the iterations for detecting boundary overlapping nodes.

2.5. Complexity Analysis

Suppose there are n_c communities and n nodes. Each community has C_i nodes in the initial partition. In the worst case, all communities connect to each other and all nodes are boundary nodes between communities. Thus, a pair of communities needs to compute $(C_i+C_j)(C_i+C_j-1)/2$ times for one iteration, and there are $n_c(n_c-1)/2$ community pairs that need to be traversed. The whole computational complexity is $O(tn_cn^2)$ in the worst case. Here t stands for the average number of iterations for each pair of communities. Since there is no oscillation, in each iteration at least one overlapping node will be identified. So in the worst case, t is equal to n, which is the number of nodes of a pair of communities containing all nodes of the network. Thus, the worst case complexity is $O(n_cn^3)$. However, in the real situation of complex networks, it will be much better for the existence of the following statistical properties of complex networks.

In the first place, a community is a kind of relatively independent structure simply because nodes connect sparsely between communities. So in most cases, the number of boundary nodes will be much smaller than the total number of nodes in the two communities. In the second place, when CONA identifies an overlapping node, the boundary node set will be updated, and many other nodes will be updated to non-boundary nodes. This possibility will bring a further reduction in the computational complexity. So this procedure will always converge in a very small number of iterations.

In addition, the statistical characteristics of complex networks show that communities always only connect to a few nearby communities [17]. Thus, the number of community pairs needed to be traversed will be much smaller than $n_c(n_c-1)/2$. That is the reason why we construct the community graph at the beginning of CONA.

Overall, CONA is very fast in real situations. The experiments in following section will illustrate this point in detail.

Table 1: Parameters of LFR synthetic networks for Fig.5 and Fig.6.

Para	Fig.5(a)/6(a)	Fig.5(b)/6(b)	Fig.5(c)/6(c)	Fig.5(d)/6(d)	Fig.5(e)/Fig.6(e)	Fig.5(f)/Fig.6(f)
N	1,000	1,000	1,000	1,000	**5,000**	1,000
k	10	10	**30**	10	10	10
k_{max}	30	30	**90**	30	30	30
c_{min}	10	**20**	10	10	10	10
c_{max}	50	**100**	50	50	50	50
μ	0.1/0.3	0.1/0.3	0.1/0.3	0.1/0.3	0.1/0.3	0.1/0.3
O_n	100	100	100	**500**	100	100
O_m	2	2	2	2	2	**4**

Table 2: Parameters of LFR synthetic networks for Fig.7-10.

Para	Fig.7	Fig.8	Fig.9	Fig.10
N	1,000	**1,000-100,000**	**1,000-1,000,000**	5,000
k	**6/12**	20	6	**20-200**
k_{max}	30	80-200	30-300	200
c_{min}	**10/20**	20	10-100	k
c_{max}	**50/100**	200-1000	100-1,000	500
μ	0.1-0.9	0.1/0.3	0.15	0.2
O_n	100	$\frac{N}{10}$	100-1,000	500
O_m	2	2	2	2

3. Results

3.1. Methodology

Currently there is almost no real-world benchmark data set for overlapping community detection. Therefore, it is rather difficult to evaluate the quality of overlapping communities accurately. In this paper, we adopt two common ways to estimate the quality of the results in experiments.

First, we will test various algorithms on synthetic networks. Recently, Lancichinetti and Fortunato proposed more realistic, LFR benchmark graphs [18], which have scale-free degree and community size distributions as well as overlapping communities. On this foundation, Normalized Mutual Information (NMI) can be used to measure the similarity of the known and found communities. Lancichinetti, et. al., gave a variant of the Mutual Information measure, that is extended to handle overlapping communities [4]. We will use this measure from Ref. [4] in the experiments in section 3.2.

The other method is to run community detection algorithms on real-world networks. One problem with this method is the difficulty of evaluating the found communities using NMI because we usually do not know the real communities that are present in the original networks. For real networks, we use overlap modularity to estimate the quality of the results as in [5, 19]. A high value corresponds to good solutions. Although modularity has some limitations, such as resolution limit [20], landscape problem [21], it is widely used in the area of community analysis.

The original modularity measure is defined only for non-overlapping communities, whereas some variants suitable for overlapping communities have been designed in recent years, such as EQ [16], Q_{ov} [22] and Q_c [23], which is proved to be equivalent to Q_{ov}. For Q_{ov}, the strength of the membership to each community should be given for each node. We assume that each vertex belongs equally to all of the communities of which it is a member. The f function used for computation of Q_{ov} is defined as:

$$f(x) = 2px - p, \tag{12}$$

where the value of p is 30, as suggested in Ref. [22]. The detailed definition of Q_{ov} is too long to be introduced here.

In the experiments in this paper, we use BGLL [11] and Infomap [10] to generate the initial partition. The parameters of the algorithms used to compare with CONA are as follows. For GCE algorithm, $k = 3$ and others adopt default values. For LFM algorithm, the value of α is set to 1.0. For COPRA, the value of v is set to 2. For CFinder, the value of k is also set to 3. To process large networks, OSLOM is used in its fastest mode, i.e. refining hard partitions generated by BGLL. For link-clustering method [8], threshold parameters are chosen by the loop program given by its authors. For different networks, although the best results might correspond to different values of parameters, we will not search for the best values of the parameters on different networks for all algorithms, including CONA. All implementations of these algorithms are supported by their authors. The results shown in the following experiments are the average of ten independent runs.

Because the non-overlapping community detection algorithms that we use in experiments generate high quality partitions, the original non-overlapping communities have already held relatively high scores on various quality measures. To show the validity of CONA, we add a random algorithm (RCONA) for comparison. RCONA finds all boundary nodes in the first step and then selects the overlapping nodes randomly in the boundary node set. The number of overlapping nodes is set to that identified by CONA.

3.2. Synthetic Networks

In this section, we will test CONA on various LFR synthetic networks. In the first part, experiments are designed to evaluate the behavior of the CONA algorithm by varying the threshold parameter φ. In the second and third part, given a fixed threshold parameter, CONA is compared with other algorithms on the quality of overlapping communities and the computational efficiency.

To construct LFR synthetic networks, ten parameters should be given as shown in Table 1. Here, N is the number of nodes, k is the average degree, k_{max} is the max degree, C_{min} is the size of the smallest community, C_{max} is the size of the largest community, t_1 is the degree exponent, which is equal to 2 in this paper, t_2 is the community size exponent, which is equal to 1 in this paper, μ is the mixing parameter, O_n is the number of overlapping nodes and O_m is the number of communities to which each overlapping node belongs.

3.2.1. Experiments on the threshold parameter of CONA

First, to analyze the properties of CONA, we test the effect of varying its threshold parameter on two groups of LFR synthetic networks. One group has a low mixing parameter equal to 0.1, and the other group has a high mixing parameter equal to 0.3. For each group, we give six kinds of networks, including networks with standard property, networks with large communities, networks with high density and networks with highly overlap in the number of overlapping nodes, networks with large scale and networks with highly overlap in the number of memberships.

For experiments on the first group of networks, as shown in Fig. 5, not all the values of the threshold parameter are suitable to generate high-quality overlapping communities. With φ ranging from 0.45 to 0.75, CONA can bring improvements for almost all kinds of networks in the experiments. At the same time, CONA can give the best results of all the algorithms used in this experiment in most cases, except for networks with highly overlap in number of overlapping nodes.

For experiments of the second group of networks, as shown in Fig. 6, the results are similar to those of the first group. The main difference is that the effective range of the value of the threshold parameter decreases. Still in most cases, with φ ranging from 0.55 to 0.75, CONA can give the best results for all the algorithms even for networks with highly overlap. Contrarily, RCONA causes a large decline in the quality of the results.

The above experimental results show that for networks with different community sizes, densities, scales or various extents of overlap in both number of overlapping nodes and memberships, when the threshold parameter φ ranges from 0.55 to 0.65, CONA can effectively convert the non-overlapping communities to overlapping communities, and the results will always be at least the same as or even better than those of other methods. Thus, we adopted 0.55 as a fixed experienced value of the threshold parameter for all following experiments. Although situations in real-world networks may be more complex, the results in the following experiments looks very good.

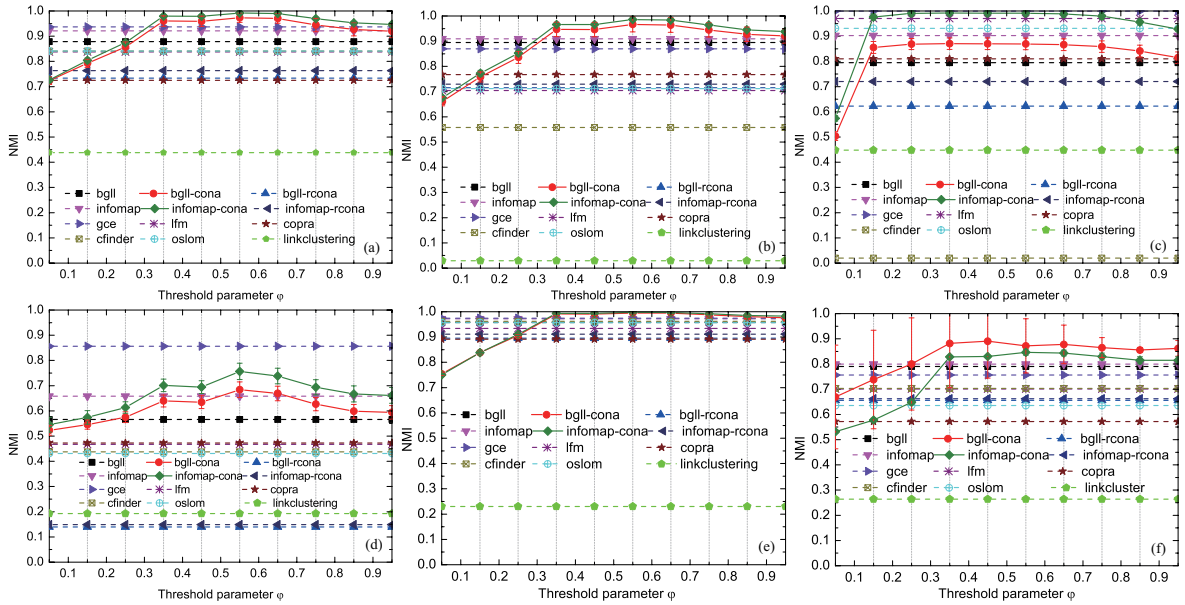

Figure 5: Experimental results on threshold parameter for networks with (a) standard property, (b) large community, (c) high density, (d) highly overlap in number of overlapping nodes, (e) large scale and (f) highly overlap in number of memberships. μ=0.1.

3.2.2. Experiments on quality

In this subsection, we test the quality of the overlapping communities generated by CONA and the other algorithms. Four groups of LFR networks are generated, including sparse networks with small communities (SS), sparse networks with large communities (SL), dense networks with small communities (DS) and dense networks with large communities (DL). For each group of networks the mixing parameter μ varies from 0.1 to 0.9. At the same time, we set one hundred overlapping nodes for each network. The addition of overlapping nodes will make the different performance of various methods appear earlier during the increase in the mixing parameter comparing with experiments on networks with disjoint communities [14, 5, 6].

In the experiment shown in Fig. 7 (a), Infomap-CONA performs best. GCE and BGLL-CONA come next. Fig. 7 (b) shows that CONA performs best in all overlapping community detection methods on sparse networks with large communities. In the results shown in Fig. 7 (c), again Infomap-CONA finds overlpping communities with the highest NMI. BGLL-CONA and GCE perform second-best in different ranges of mixing parameter, respectively. Lastly, in Fig. 7 (d), for networks with high density and large communities, CONA gives the best results once again. LFM and link-clustering method can only give poor results for all four groups of networks. For OSLOM algorithm, refining the partitions generated by the same algorithm (BGLL), it never gives better results than those of CONA. In comparison, RCONA decreases the score of NMI for all networks.

It is also very import to test the quality of results of CONA on larger networks. Fig. 8 shows the results of CONA on some larger networks. There are two groups of networks with small and large value of mixing parameter, respectively. In each group, the scales of networks are from 1,000 nodes to 100,000 nodes. At the same time, some other parameters of LFR networks are adjusted on average in the given ranges. For each network, we set the number of overlapping nodes to one tenth of number of nodes of the network. From Fig. 8 we can see that CONA can solve large networks very well. Thus we can conclude taht the network scale does not obviously affact the performance of CONA.

The above results show that CONA performs very well in producing high-quality overlapping communities. In some cases, however, when the mixing parameter $\mu > 0.5$, CONA often fails to optimize NMI. Nevertheless, in these cases, the performance of the other overlapping community detection methods also declines rapidly with an increase in the mixing parameter. Solving this problem for networks with overlapping

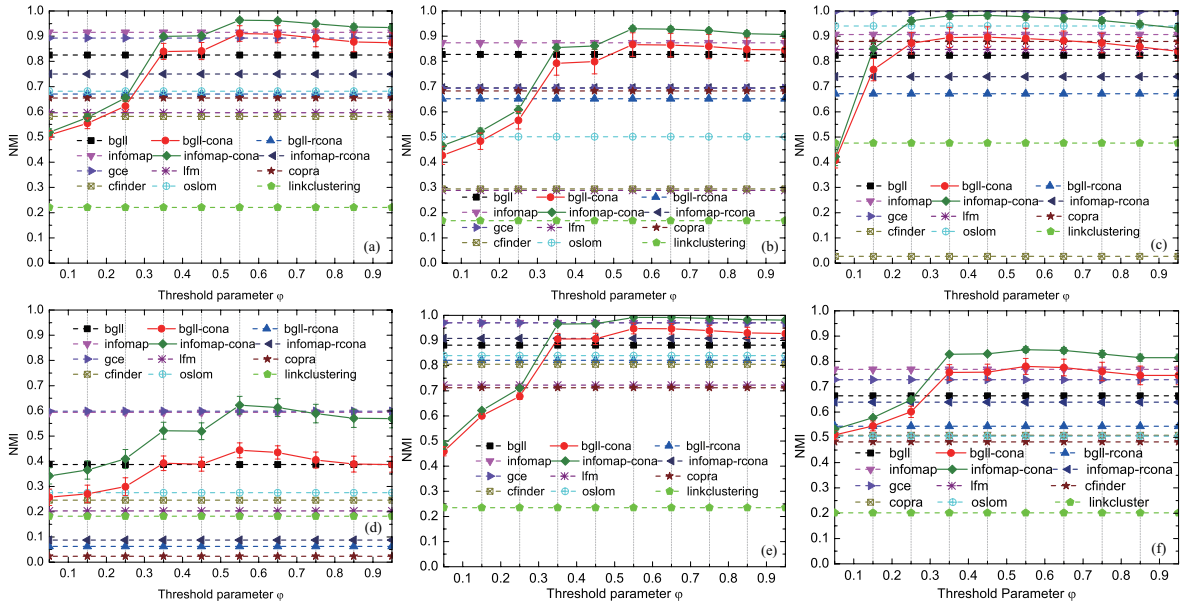

Figure 6: Experimental results on threshold parameter for networks with (a) standard property, (b) large community, (c) high density, (d) highly overlap in number of overlapping nodes, (e) large scale and (f) highly overlap in number of memberships. μ=0.3.

communities and a high value of mixing parameter might require more effort in the future.

3.2.3. Experiments on computational efficiency

Finally, we experimentally evaluate the speed of CONA on networks of different scales and densities.

In the experiment shown in Fig. 9, we vary the number of nodes N. At the same time, the max node degree k_{max}, community size and the number of overlapping nodes are adjusted on average with increasing N. As analyzed in section 2.5, CONA takes very little time. When processing networks with 1,000,000 nodes, BGLL-CONA only spends about one hundred seconds. Other two algorithms that can finish in one thousand seconds for this data are link-clustering method and GCE. Infomap can solve networks with 500,000 nodes, and based on the partition generated by Infomap, the time CONA costs can almost be ignored. The total time of Infomap-CONA is a little more than that of COPRA and less than that of CFinder. OSLOM seems to be the slowest method in this experiment.

Fig. 10 shows the experimental results from various methods for networks with different densities. The average node degree k is set from 20 to 200, and as it increases, the size of the minimum community will become larger, which is set as the value of k. We can find that CONA, BGLL and Infomap are not insensitive to the average node degree, whereas the time expenses of GCE, OSLOM and link-clustering method rise rapidly with the increment in the average node degree. The performances of the left two algorithms, COPRA and LFM, are also affected by the average node degree to some extent. Because CFinder even needs more than one thousand seconds on average for the sparsest networks, it is not compared in this experiment.

The above two experiments show that CFinder, link-clustering, OSLOM and GCE can only solve sparse large-scale networks. Though COPRA is not affected by the network density very much, it can only process networks with 500,000 nodes in 1,000 seconds. CONA combining high quality non-overlapping community detection algorithms is almost the only solution in current methods for overlapping community detection on *large-scale dense* networks.

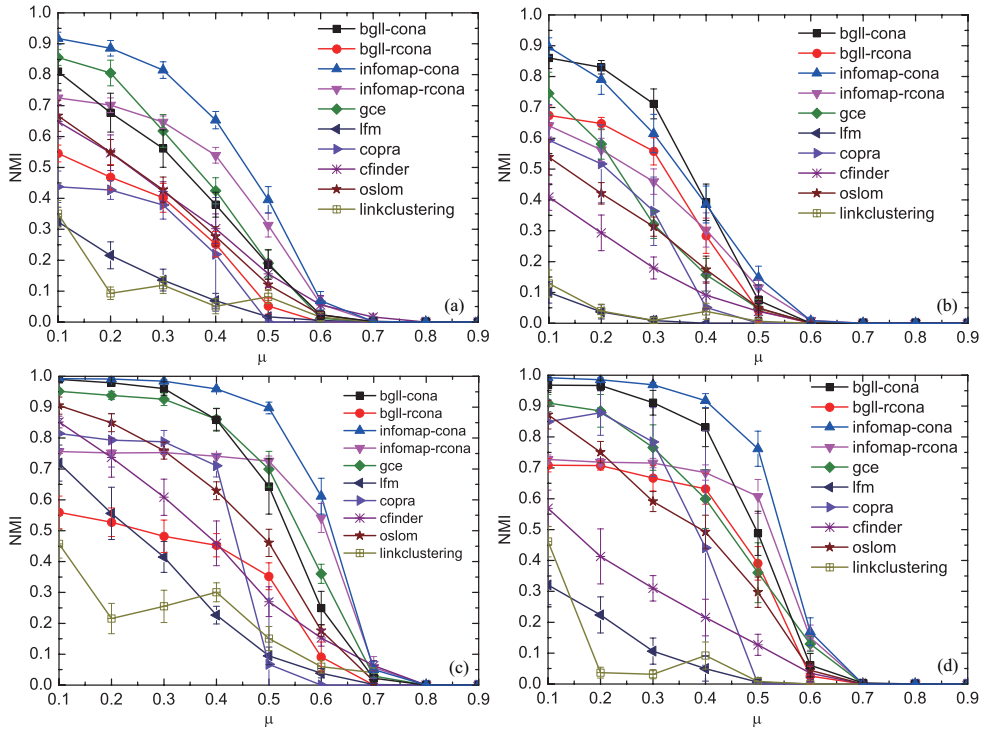

Figure 7: Comparing NMI on (a) SS network, (b) SL network, (c) DS network and (d) DL network.

3.3. Real-World Networks

3.3.1. Experiments on small networks

The **karate** network [24] is a small benchmark data set, and has been widely used to test community detection algorithms. This network describes the relationships between persons of a karate club that contains 34 persons and 78 relations and is assembled by Zachary . Due to a contrast between one of the instructors and the club administrator, the club separated into two groups, that are split up by a dashed line in Fig. 11 and Fig. 12.

BGLL finds four communities in the karate network as shown in Fig. 11 (a). If we recogonized the left two small communities and right two small communities as two larger communities, the result is almost correct except for the community membership of node 10. Based on the partition generated by BGLL, CONA discoveries six overlapping nodes among communities, as shown in Fig. 11 (b). From the result we

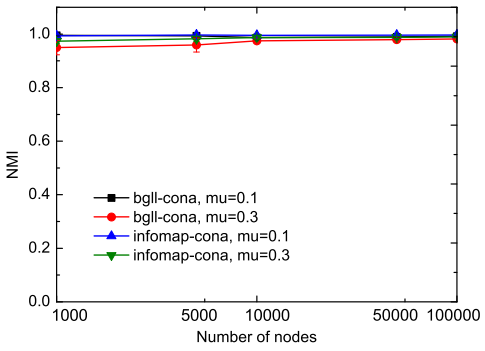

Figure 8: Quality test of CONA on LFR networks with larger scales.

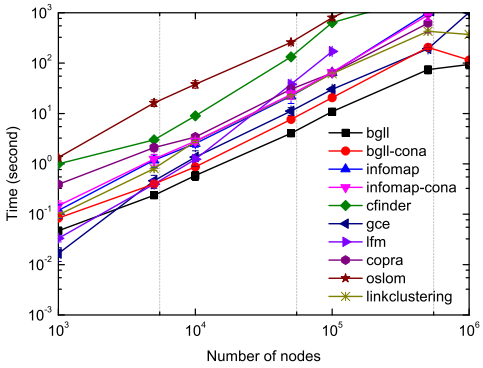

Figure 9: Runtime results of various methods for networks with different scales.

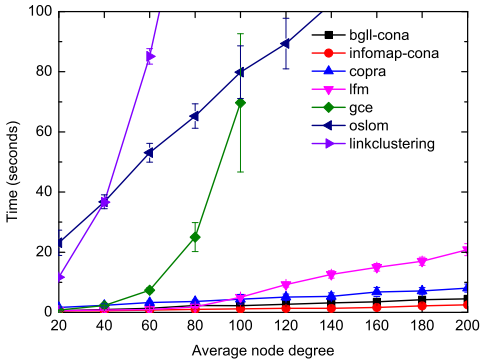

Figure 10: Runtime results of various methods for networks with different density.

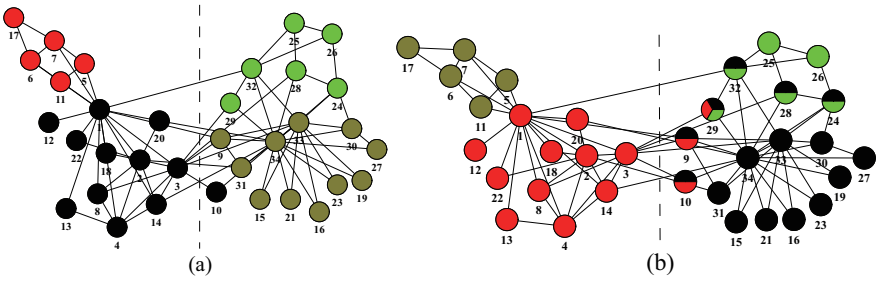

Figure 11: Application of (a) BGLL and (b) BGLL-CONA to karate network.

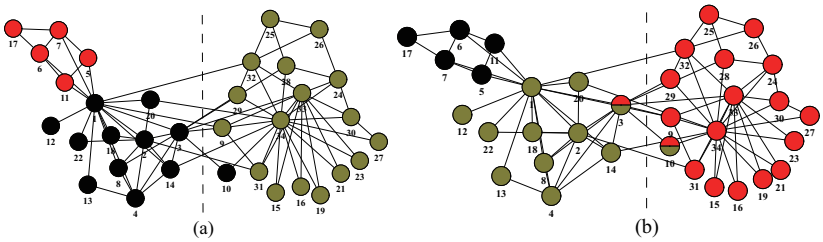

Figure 12: Application of (a) Infomap and (b) Infomap-CONA to karate network.

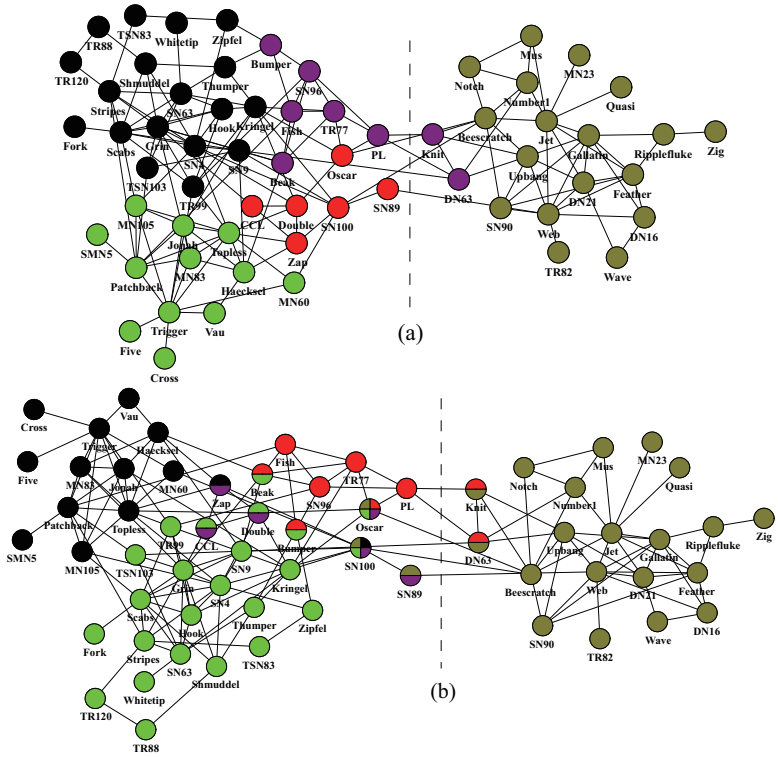

Figure 13: Application of (a) BGLL and (b) BGLL-CONA to dolphins network.

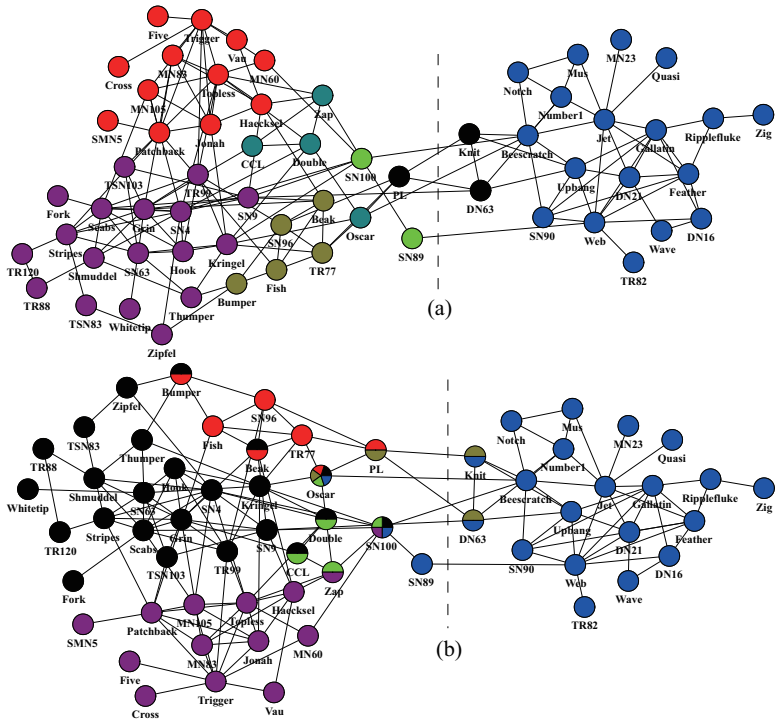

Figure 14: Application of (a) Infomap and (b) Infomap-CONA to dolphins network.

Table 3: Experimental results for the Ca-hepPh network.

algorithms	time	Q_{ov}	EQ
BGLL-CONA	1.33s	0.78	0.63
BGLL-RCONA	1.06s	0.581	0.461
Infomap-CONA	3.83s	0.71	0.581
Infomap-RCONA	3.31s	0.484	0.4
GCE	8.92s	0.524	0.492
COPRA	7.51s	0.538	0.271
LFM	1.37s	0.271	0.268
OSLOM	105.4s	0.589	0.524

can see that node 10, which is misclassified by BGLL, becomes an overlapping node. It means that CONA fixes a mistake of BGLL, and detects overlapping nodes successfully.

Fig. 12 (a) shows the communities detected by Infomap algorithm. Same with communities detected by BGLL, most nodes have correct community membership except node 10. Under this partition, nodes 3 and 10 are identified as overlapping nodes by CONA. Once again, CONA corrects a mistake made by a non-overlapping community detection algorithm and identifies two overlapping nodes successfully.

The **dolphins** network [25] is another classical benchmark data set. It is a social network of a community of 62 bottlenose dolphins living in Doubtful Sound, New Zealand. The network was compiled by Lusseau from seven years of field studies of dolphins, with ties between dolphins pairs being established by observation of statistically significant frequent association. The two actual communities analyzed by Lusseau are split up by a dashed line in Fig. 13 and Fig. 14.

BGLL finds five communities in the dolphins network as shown in Fig. 13 (a). In the result, two node, "Knit" and "DN63" are misclassified by BGLL. Fig. 13 (b) shows the result generated by CONA based on the partition of Fig. 13 (a). As is hoped, node "Knit" and "DN63" are successfully identified as overlapping nodes between the two communities split by the dashed line.

Fig.14 (a) shows that Infomap detects seven communities in the dolphins network, and there are some very small communities in the results. A small community containing node "PL", "Knit" and "DN63" is on the border of the two real communities. Regardless of which side it is divided to, there will be some misclassified nodes. CONA fixes the result through identifying them as overlapping nodes, as shown in Fig. 14 (b). Although node "SN89" is misclassified to the other community by CONA, it is hard to say which community it should belong to. Because node "SN89" has only two links connecting to different communities.

Through the above experiments on small networks, we can conclude that CONA can not only identify overlapping nodes between communities, but correct some mistakes made by non-overlapping community detection algorithms.

3.3.2. Experiments on large real-world networks

Although many methods can solve large data sets as shown in the last subsection, synthetic networks might not share all the properties of real networks. Here, we will test the method proposed in this paper on five large real-world networks, and compare the runtime and modularity measures with other four relatively fast algorithms: GCE, COPRA, LFM and OSLOM.

The **Ca-hepPh** data set is a collaboration network from the e-print arXiv and covers scientific collaborations between authors of papers submitted to the High Energy Physics - Phenomenology category [26]. The network contains 12,008 nodes and 237,010 edges. Table 3 shows the results of various methods for this network. CONA gives the best results in terms of quality out of all the methods. Although LFM takes as little time as BGLL-CONA, it can only produce communities with low modularity. Other methods only detects communities of middling quality with costing more time than CONA.

The **Enron** email communication network covers all the email communications within a data set of around a half million emails [27]. The nodes of the network are email addresses, and there is an edge

Table 4: Experimental results on Enron network.

algorithms	time	Q_{ov}	EQ
BGLL-CONA	4.51s	0.74	0.559
BGLL-RCONA	2.59s	0.579	0.421
Infomap-CONA	27.78s	0.558	0.535
Infomap-RCONA	25.01s	0.35	0.325
GCE	348s	0.38	0.397
COPRA	11.8s	0.704	0.315
LFM	8.64s	0.196	0.196
OSLOM	342.22s	0.274	0.294

Table 5: Experimental results for the Facebook network.

algorithms	time	Q_{ov}	EQ
BGLL-CONA	20.76s	0.773	0.589
BGLL-RCONA	8.2s	0.593	0.426
Infomap-CONA	232.1s	0.514	0.502
Infomap-RCONA	207s	0.306	0.289
GCE	4,057s	0.272	0.298
COPRA	101.7s	0.797	0.558
LFM	59.4s	0.076	0.077
OSLOM	1029.1s	0.428	0.365

between two nodes if at least one email exists between them. Lastly, this network consists of 36,692 nodes and 367,662 edges. In Table 4, we compare the results of CONA and other methods for the Enron email network. BGLL-CONA finds the best result with the least time. GCE and OSLOM attain relatively low score of modularity with costing much more time than other methods. Both COPRA and LFM cost about 10 seconds, but LFM only gives very poor results and COPRA gets two different results on Q_{ov} and EQ, respectively. On Q_{ov} , COPRA gets the second best result, whereas on EQ it gets the second worst result.

The **Facebook** data set [28] is a user-to-user friendship network from the Facebook New Orleans networks. It contains 63,731 nodes and 817,090 undirected edges. The results on this network are shown in Table 5. BGLL-CONA produces almost the best result, which is similar to that of COPRA, whereas only using one fifth of time of COPRA. GCE takes the longest time and only gives poor result. Although LFM runs comparatively fast, the quality of its result is very low. The runtime of Infomap-CONA on this data set is a bit long, whereas the quality of the result is not bad.

The **Web-stanford** [29] is a data set of webpages and hyperlinks between webpages from Stanford University. The directed network has been symmetrized, and loops have been removed. The original network has been transformed into a network with 281,903 nodes and 1,992,636 undirected edges. As shown in Table

Table 6: Experimental results for the Stanford network.

algorithms	time	Q_{ov}	EQ
BGLL-CONA	225.3s	0.977	0.925
BGLL-RCONA	220.4s	0.651	0.624
Infomap-CONA	609.3s	0.894	0.865
Infomap-RCONA	593.9s	0.756	0.736
GCE	$> 24h$	–	–
COPRA	440.95s	0.397	0.397
LFM	2,383s	0.28	0.28
OSLOM	16823s	0.59	0.597

Table 7: Experimental results for the mobile network.

algorithms	time	Q_{ov}	EQ
BGLL-CONA	387.9s	0.587	0.499
BGLL-RCONA	89.5s	0.401	0.321
Infomap-CONA	18,991s	0.361	0.38
Infomap-RCONA	18,732s	0.212	0.21
GCE	2,541s	0.101	0.141
COPRA	404.9	0.036	0.013
LFM	4,736s	0.047	0.053
OSLOM	9138.8s	0.205	0.21

Table 8: Results for huge real-world networks. *no* stands for the number of overlapping nodes. *avm* stands for the average number of memberships. *nc* stands for the number of communities. *avsc* stands for the average size of communities.

networks	algorithms	time	*no*	*avm*	*nc*	*avsc*
Youtube	BGLL-CONA	2366.5s	98736	1.096	10553	118.2
Youtube	OSLOM	71142s	2756	1.003	20246	15.7
Youtube	COPRA	2,302s	9011	1.008	12393	56.1
As-skitter	BGLL-CONA	905.9s	71815	1.044	2531	699.9
Flickr	BGLL-CONA	16329.3s	231685	1.145	45527	43.15
Livejournal-1	BGLL-CONA	24150.8s	580466	1.139	8515	648.26

6, CONA gives the best results. BGLL-CONA takes the least time to achieve the best modularity. Although COPRA runs slightly faster than Infomap-CONA, it only detects low quality communities. Again, LFM produces a poor-quality result. OSLOM gets results with middling quality on overlap modularity, while it takes too much time. For this network, GCE cannot finish within one day.

We obtained an anonymous call relation data set for a small town from one of the largest mobile communication service providers in China. This **mobile** social network contains 348,808 users that are identified by a random unique number and 3,644,779 call relations between them, which are collected and summed over one month. As shown in Table 7, BGLL-CONA gives the best results both for speed and quality. Infomap-CONA gives the second-best result for quality taking several hours. All other methods can only produce results with very small score on modularity.

3.3.3. Experiments on huge real-world networks

Here, we also challenge these methods using four huge real-world networks. The Youtube data set [30] is a friendship network that contains 1,138,499 users and 2,990,443 relations between the users. The As-skitter data set [27] is a large-scale Internet topology graph with 1,696,415 nodes and 11,095,298 edges. The Flickr friendship network [30] contains 1,715,255 nodes and 15,555,041 edges. The Livejournal network [29] contains 4,847,571 nodes and 43,110,428 edges.

On these huge networks, most current methods do not work. Table 8 lists the successful records, among which only two records are generated by other methods. CONA can solve all these four huge networks in one day. Since the computataional complexity of overlap modulariy is too high for huge networks, here we give some other statistics, such as *no*, representing the number of overlapping nodes, *avm*, representing the average number of memberships, *nc*, standing for the number of communities and *avsc*, which stands for the average size of communities. From the results on Youtube dataset generated by the three algorithms, we find that CONA detects much more overlapping nodes than COPRA and OSLOM. The result generated by CONA has larger average number of memberships and larger average size of communities. Since the average size of communities of the result generated by COPRA is comparatively small, it detects more communities than other algorithms.

All the above results show that CONA offers a considerable speed advantage, especially for huge real-world networks. At the same time, the quality of overlapping communities produced by CONA is always at

least similar, or even better than those of other methods.

4. CONCLUSIONS

In this paper, we presented an efficient algorithm, CONA, to detect overlapping communities in large-scale networks starting from a high quality partition. Based on the deduced conditions for overlapping nodes, the proposed algorithm identifies overlapping nodes from the boundary and inner node set in turn. On the aspect of quality, CONA can always give better results than the other algorithms used in this paper. On the aspect of speed, the proposed method performs very well, especially for huge real-world networks.

An advantage of CONA is that it is easy to extend to weighted networks by replacing degree with sum of link-weights. In addition, the algorithm is highly amenable to parallel implementation because discovering overlapping nodes between communities of different pairs is completely independent.

From the results in Table. 8, we can find that there is almost no method that can detect highly overlapping communities in very large networks. GCE algorithm is good at detecting highly overlapping communities, while it can not process very large real-world networks. Therefore, detecting highly overlapping communities in very large real-world networks is still a hard problem to be solved in the future work.

5. ACKNOWLEGEMENTS

This work was supported by the Beijing Natural Science Foundation (Grant No. 4112046). The authors thank Professor M.E.J. Newman and Assistant Professor J. Leskovec for providing the network data sets. We thank Professor S. Gregory; G. Palla, Ph.D.; A. Lancichinetti, Ph.D.; and C. Lee, Ph.D. for providing the source code for their algorithms.

References

[1] M. E. J. Newman, The structure and function of complex networks, SIAM Review 45 (2003) 167.
[2] S. Fortunato, Community detection in graphs, Physics Reports 486 (2010) 75–174.
[3] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[4] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New JOURNAL of Physics 11 (2009) 033015.
[5] S. Gregory, Finding overlapping communities in networks by label propagation, New JOURNAL of Physics 12 (2010) 103018.
[6] C. Lee, F. Reid, A. McDaid, N. Hurley, Detecting highly overlapping community structure by greedy clique expansion (2010).
[7] A. Lancichinetti, F. Radicchi, J. Ramasco, Finding statistically significant communities in networks, PloS one 6 (2011) e18961.
[8] Y. Ahn, J. Bagrow, S. Lehmann, Link communities reveal multiscale complexity in networks, Nature 466 (2010) 761–764.
[9] M. Girvan, M. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences of the United States of America 99 (2002) 7821.
[10] M. Rosvall, C. Bergstrom, Maps of random walks on complex networks reveal community structure, Proceedings of the National Academy of Sciences 105 (2008) 1118.
[11] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, JOURNAL of Statistical Mechanics: Theory and Experiment 2008 (2008) P10008.
[12] U. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical Review E 76 (2007) 36106.
[13] M. Ovelgónne, A. Geyer-Schulz, M. Stein, Randomized greedy modularity optimization for group detection in huge social networks, in: SNA-KDD10: Proceedings of the 4th Workshop on Social Network Mining and Analysis.
[14] A. Lancichinetti, S. Fortunato, Community detection algorithms: a comparative analysis, Physical Review E 80 (2009) 56117.
[15] M. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical review E 69 (2004) 26113.
[16] H. Shen, X. Cheng, K. Cai, M. Hu, Detect overlapping and hierarchical community structure in networks, Physica A: Statistical Mechanics and its Applications 388 (2009) 1706–1712.
[17] S. Muff, F. Rao, A. Caflisch, Local modularity measure for network clusterizations, Physical Review E 72 (2005) 56107.
[18] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Physical Review E 80 (2009) 16118.

[19] D. Chen, M. Shang, Z. Lv, Y. Fu, Detecting overlapping communities of weighted networks via local algorithm, Physica A: Statistical Mechanics and its Applications (2010).
[20] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences 104 (2007) 36.
[21] B. Good, Y. De Montjoye, A. Clauset, Performance of modularity maximization in practical contexts, Physical Review E 81 (2010) 046106.
[22] V. Nicosia, G. Mangioni, V. Carchiolo, M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities, JOURNAL of Statistical Mechanics: Theory and Experiment 2009 (2009) P03024.
[23] H. Shen, X. Cheng, J. Guo, Quantifying and identifying the overlapping community structure in networks, Journal of Statistical Mechanics: Theory and Experiment 2009 (2009) P07042.
[24] W. Zachary, An information flow model for conflict and fission in small groups, JOURNAL of Anthropological Research 33 (1977) 452–473.
[25] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, S. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behavioral Ecology and Sociobiology 54 (2003) 396–405.
[26] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (2007) 2.
[27] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 177–187.
[28] B. Viswanath, A. Mislove, M. Cha, K. Gummadi, On the evolution of user interaction in facebook, in: Proceedings of the 2nd ACM workshop on Online social networks, pp. 37–42.
[29] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, Internet Mathematics 6 (2009) 29–123.
[30] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and analysis of online social networks, in: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 29–42.