A Fast Overlapping Community Detection Algorithm Based on Weak Cliques for Large-Scale Networks

Xingyi Zhang, Congtao Wang, Yansen Su, Linqiang Pan, and Hai-Feng Zhang

***Abstract*— Community detection is an important tool to analyze hidden information such as functional module and topology structure in complex networks. Compared with traditional community detection, it is more challenging to find overlapping communities in complex networks, especially when the networks are of large scales. Among various overlapping community detection techniques, the well-known clique percolation method (CPM) has shown promising performance in terms of quality of found communities, but suffers from serious curse of dimensionality due to its high computational complexity, which makes it very unlikely to be applied to large-scale networks. To address this issue, in this paper, we propose a weak-CPM for overlapping community detection in large-scale networks. A new measure for characterizing the similarity between weak cliques is also suggested to check whether the weak cliques can be merged into a community. Experimental results on synthetic and real-world networks demonstrate the competitive performance of the proposed method over six popular overlapping community detection algorithms in terms of both computational efficiency and quality of found communities. In addition, the proposed method is also suitable for detecting large-scale networks with an unclear community structure under different levels of overlapping density and overlapping diversity, which is an important property of many real-world complex networks.**

***Index Terms*— Clique percolation, complex network, large scale, overlapping community, weak clique.**

I. Introduction

SINCE most complex systems such as social networks, biological networks, and web networks can be naturally formulated as complex networks [1]–[5], researchers have proposed various strategies for analyzing complex networks. As one of the most important complex network analysis strategies, community detection is often applied to mine hidden information in complex networks, such as the functional module and topology structure, most of which are not easy to be identified by empirical observations.

Generally speaking, communities in a network refer to groups of nodes such that links between nodes in the same group are dense, whereas links between nodes in different groups are sparse [6]–[8]. In the past few years, a large number of community detection techniques have been developed, e.g., hierarchical clustering [9], [10], spectral clustering [11], [12], random walks [13]–[15], and optimization-based algorithms [16]–[18], most of which focus on finding disjoint communities in complex networks. As a consequence, the communities detected by these techniques satisfy the restriction that each node should belong to one and only one community. In many real-world networks, however, this restriction cannot always be satisfied, i.e., some nodes can belong to more than one community. Such kind of communities that contain overlapping nodes are usually called overlapping communities [19].

There are various instances of overlapping communities that can be found in real-world scenarios. For example, in the collaboration network, a scientist can belong to several communities if he/she collaborates with others in different fields [20]; a single protein may be part of more than one protein complex, hence overlapping community detection in protein interaction networks will promote us to find proteins having different functions [21]; in the immune system, cells can be divided into many populations based on the specific combinations of cell surface markers or broader functional families, such as macrophages, natural killer cells, and dendritic cells, so overlapping community detection will help to reveal the different types of cells [22].

Due to the wide existence of overlapping communities, it becomes particularly important to design corresponding community detection techniques. As a representative work, Palla *et al.* [23] proposed a clique percolation method, termed CPM. The CPM was shown to be very suitable for detecting overlapping communities as it can correctly identify the most overlapping communities in complex networks, even when both overlapping density and overlapping diversity are high [19], [24], [25]. In spite of the promising performance of CPM, however, the algorithm suffers from an expensive computational cost, which makes it not very likely (if not impossible) to be applied to large-scale complex networks. To address this issue, this paper presents a weak-CPM, termed W-CPM, for overlapping community detection in large-scale complex networks. To summarize, the main contributions of the work are as follows.

Manuscript received February 8, 2016; revised August 12, 2016 and May 10, 2017; accepted September 1, 2017. Date of publication September 21, 2017; date of current version November 21, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61672033, Grant 61502004, and Grant 61502001. *(Corresponding author: Yansen Su.)*

X. Zhang, C. Wang, and Y. Su are with the Institute of Bio-inspired Intelligence and Mining Knowledge, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: xyzhanghust@gmail.com; wctahuedu@163.com; suyansen1985@163.com).

L. Pan is with the Key Laboratory of Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: lqpan@mail.hust.edu.cn).

H.-F. Zhang is with the School of Mathematical Science, Anhui University, Hefei 230039, China (e-mail: haifengzhang1978@gmail.com).

Digital Object Identifier 10.1109/TCSS.2017.2749282

1) We propose a W-CPM to address the high computational cost of CPM for overlapping community detection in large-scale complex networks. In W-CPM, we do not exactly identify the k-cliques but only find some weak cliques determined by two nodes in the network. In identifying a weak clique between two nodes, we only need to check the common neighbors of the two nodes. Therefore, the identification of weak cliques developed in W-CPM is much more efficient than that of k-cliques adopted in a variety of existing methods based on clique percolation theory.
2) We suggest a new measure in W-CPM to evaluate the similarity between weak cliques for determining whether the weak cliques we find can be merged into a community or not. The suggested measure considers not only the sharing nodes but also the links between weak cliques, which enables the W-CPM to detect communities in networks without a clear structure since these communities usually consist of several weak cliques that have a small number of common nodes.
3) Experimental results on synthetic and real-world networks demonstrate the competitive performance of the proposed W-CPM for large-scale complex networks in terms of both computational efficiency and solution quality, especially when the community structure of networks is not clear.

The rest of this paper is organized as follows. In Section II, we briefly recall the related work on existing overlapping community detection algorithms. The details of the proposed W-CPM are described in Section III. Empirical results by comparing W-CPM with six state-of-the-art methods are presented in Section IV. Finally, conclusions and future work are given in Section V.

II. Related Work

In the past 10 years, overlapping community detection has been recognized as one of the hot areas in complex networks and a variety of algorithms have been developed to address overlapping community detection problem, which can be largely divided into four categories. The first group of algorithms uses the local expansion and optimization, where a community is found by maximizing a local fitness function from a seed in the network to be detected. Hence, the seed and the local fitness function are two key issues for designing local expansion and optimization-based algorithms. Baumes *et al.* [26] suggested to use disjoint cores of clusters as seeds, and expand these seeds by adding or removing nodes until the ratio of links within them to those outside arrives at the maximum. Lancichinetti *et al.* [27] proposed a local expansion and optimization algorithm, termed LFM, by adopting a random node as the seed. The LFM is the first algorithm that can find both overlapping communities and the hierarchical structure. Lee *et al.* [28] developed a greedy clique expansion, called GCE, for overlapping community detection, where distinct cliques were adopted as seeds. Empirical results demonstrate that the GCE holds a competitive performance in detecting high overlapping communities. Recently, Bandyopadhyay *et al.* [29] developed a fast overlapping community search (FOCS) method, which was also based on the idea of local expansion and optimization. The FOCS is suited to detect overlapping communities in large-scale networks, since its time complexity is linear to the number of links and nodes of the network.

The second group of algorithms is based on the label propagation, which has been widely investigated in detecting communities without any overlap [30]–[33]. For overlapping community detection, the label propagation allows a node in the network to have multilabels during the process of propagating the labels. Two representative label propagation algorithms for overlapping community detection are community overlap propagation algorithm (COPRA) [34] and speaker-listener label propagation algorithm (SLPA) [35]. The COPRA achieves multiple labels of a node by defining the belonging coefficients of this node according to the number of same labels in its neighbors, while the SLPA accumulates knowledge of repeatedly observed labels of each node by mimicking human communication behavior to have multiple labels. It has been shown that the most community detection algorithms based on label propagation hold a low computational cost, but they often fail to find the communities with a small size even when these communities are well connected [29].

The third group of algorithms partition links, instead of nodes, into communities. With the link communities, a node is overlapping if the links connected with the node are assigned to more than one cluster. Hence, link clustering converts the overlapping community detection into a disjoint clustering problem [36]–[38]. The idea of link community was first suggested by Ahn *et al.* [39] and a link community clustering algorithm, termed LC, was proposed based on link similarity. Similar idea has also been considered in [40], where Evans and Lambiotte suggested to convert a network into a link graph whose nodes are the links of the original network. In this way, existing disjoint community detection algorithms can be directly applied to detect overlapping communities. Although the link community is an interesting idea for overlapping community detection, it is easy to produce high overlapping results since a node is often connected to links that are partitioned to many different clusters [41].

The fourth group of algorithms uses the clique percolation theory, where a community is considered to consist of several small, fully connected subgraphs that share nodes. The first clique percolation theory-based algorithm for detecting overlapping communities, termed CPM, was suggested by Palla *et al.* in 2005 [23]. Since then, CPM has drawn much attention from researchers in different fields due to the fact that it can correctly identify the most overlapping communities in complex networks, even when the overlapping density and overlapping diversity are high [19], [24], [25]. Although the performance of CPM is promising, it suffers from the high computational cost, which makes it very unlikely to be applied to large-scale complex networks. There are two main reasons for the inefficiency of CPM. First, all cliques with a size of no less than k (a clique with a size of k is called k-clique) need to be found in a network, which is an nondeterministic polynomial time (NP)-complete problem [42]. Second, each pair of cliques is checked to determine whether they can be merged into a larger community, which is also very time consuming.

To improve the computational efficiency of CPM, Kumpula *et al.* [43] proposed a sequential algorithm for fast clique percolation in 2008, termed SCP. The main idea of SCP is to decrement the number of comparisons between cliques when determining whether the found cliques can be merged into larger communities. This is achieved by sequentially inserting constituent links to the network for identifying k-cliques and simultaneously keeping track of the emerging community structure by constructing a bipartite network which consists of k-cliques and $(k-1)$-cliques. In this way, SCP only needs to find each pair of k-cliques sharing a $(k-1)$-clique in the bipartite network, and they will be merged into a community. Empirical results have illustrated that SCP is more efficient than CPM when the value of k is relatively small, whereas its efficiency will considerably decrease as the value of k increases [43]. Reid *et al.* [44] developed a fast clique percolation algorithm for community detection in 2012, where the minimal spanning tree was adopted to reduce the number of comparisons between cliques. It is necessary to note that both SCP and Reid's algorithm focused on performing as few comparisons between cliques as possible to improve efficiency. For a given network to be detected, however, they both need to identify all k-cliques in the network, which consumes most of the runtime of the known clique percolation algorithms in community detection.

In this paper, we propose a fast clique percolation algorithm W-CPM for overlapping community detection, which focuses on reducing the computational cost of identifying k-cliques in the network. The main idea of the proposed algorithm is that we do not exactly identify the k-cliques but only find some weak cliques in the network. It is worth noting that there exist some relaxed definitions of clique, such as k-core [45] and k-dense core [46], however, they are different from the weak clique suggested here. To reduce the computational cost, the proposed weak clique is determined by key nodes in the network, whereas the identification of k-core and k-dense core is still very time consuming despite that their definitions are relaxed. The details of W-CPM will be presented in the next section. We should stress that there are also many overlapping community detection algorithms which adopt different ideas from those discussed above, e.g., community-affiliation graph model [47], nonnegative matrix factorization [48], [49], and evolutionary computation [50]–[53].

III. Proposed Algorithm

The driving principle for W-CPM is that a community often consists of several weak cliques instead of cliques in complex networks, especially when the network does not have a clear community structure. The key idea of W-CPM is to find weak cliques in a network and then merge the weak cliques to obtain larger communities. The general framework of W-CPM is presented in Algorithm 1, which consists of two main steps: 1) identifying the weak cliques in the network and 2) merging these weak cliques into communities based on the suggested similarity measure between weak cliques. In the following, we describe in detail the above two steps.

Algorithm 1 General Framework of W-CPM

```
Input: Network $G(V, E)$, threshold $T$
Output: S={$S_i$ | $S_i \subseteq$ V  and  $S_i$  is  a  community }
$S \leftarrow \emptyset$
$WClique \leftarrow \boldsymbol{IdentifyWeakClique}(G)$
/*all the weak cliques initially unvisited*/
for each wclique ∈ WClique do
    mark wclique as unvisited
end for
/*if weak clique $i$ is not visited, merge it with its neighboring weak cliques*/
for each $wclique \in WClique$ do
    if $wclique$ is unvisited then
        $\boldsymbol{Merge(wclique, S, T)}$
    end if
end for
return $S$
```

A. Identifying Weak Cliques in Complex Networks

It is known that finding all cliques in a network is an NP-complete problem [42], which makes the community detection algorithms based on cliques very time consuming. To reduce the computational cost, we propose the definition of weak cliques determined by key nodes. We define a weak clique determined by two adjacent nodes as the subgraph consisting of the two nodes and all neighbors shared by them, which is computationally very efficient once the two nodes are obtained. For given networks or graphs, the weak cliques obtained here are a subset of the weak cliques according to graph theory [54], where all subgraphs satisfying that the distance between every two nodes in the subgraphs is at most two are defined as weak cliques. In this paper, the weak cliques are obtained with the restriction that they are determined by two adjacent nodes. Formally, a weak clique determined by two adjacent nodes is defined as follows.

Definition 1 (Weak Clique Determined by Two Adjacent Nodes): Given an undirected network $G = (V, E)$, where V is the set of nodes and E is the set of links. Let u and v be two adjacent nodes in G, then the weak clique determined by u and v is defined as

$$G_{uv} = (V_{uv}, E_{uv}) \tag{1}$$

where $V_{uv} = \{u, v\} \cup (N_u \cap N_v)$, $E_{uv} = \{(x, y) \in E \mid x, y \in V_{uv}\}$, $N_u = \{x \mid (u, x) \in E\}$, and $N_v = \{x \mid (v, x) \in E\}$.

As shown in Fig. 1, the weak clique determined by adjacent nodes a and c is the subgraph consisting of nodes a, b, c, e, and f. It is not difficult to find that the node f and the other nodes of the weak clique determined by a and c do not belong to the same community. So, we can conclude that not all nodes of a weak clique defined above belong to the same community, which is closely related to the two adjacent nodes that determine the weak clique. This means that we should carefully choose the two adjacent nodes such that all nodes of the obtained weak clique belong to the same community as much as possible. For this reason, in the following, we propose the definition of priority of node.

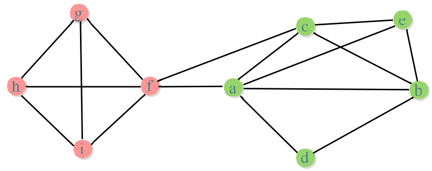


Fig. 1. Illustrative example of weak cliques. This network has two communities, one consisting of nodes g, f, h, and i and the other consisting of a, b, c, d, and e. In this network, the weak clique determined by nodes a and c consists of nodes a, b, c, e, and f.

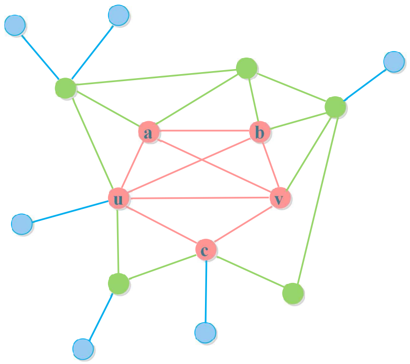


Fig. 2. Example to illustrate the reason why we choose the node with a maximum priority and the one with a maximum similarity with it to determine a weak clique. The node u has the maximum priority, node v has the maximum similarity with u, and the weak clique determined by u and v consists of nodes u, v, a, b, and c.

Definition 2 (Priority of Node): Given an undirected network $G = (V, E)$, where V is the set of nodes and E is the set of links. Let u be a node in G, then the priority of node u is defined as

$$P_u = \frac{m_u + k}{k + 1} \tag{2}$$

where m_u is the number of links between the neighbors of u and k is the degree of node u.

From the above definition, we can find that the priority of a node in a network is a centrality measure, which aims to estimate the strength of links in the neighborhood of the node. Note that the priority of node presented here is different from the degree centrality and local clustering coefficient. For a node in a network, only the degree of the node is considered in degree centrality [55] and the number of links between its neighbors in local clustering coefficient [56], whereas both degree of the node and the number of links between its neighbors are included in priority of node, which helps to estimate better the strength of links in the neighborhood of the node. The larger the priority of a node, the denser the connections in the neighborhood of the node. We therefore choose the node that has a maximum priority in the network to determine weak cliques. The other node that we choose to determine weak cliques is the adjacent one that has a maximum similarity with the node having a maximum priority. The adopted similarity between nodes is a measure that estimates the possibility of two nodes belonging to the same community, which has been widely used in complex networks [57], [58]. To this end, we introduce the definition of Salton index developed in [59].

Definition 3 (Salton Index [59]): Given an undirected network $G = (V, E)$, where V is the set of nodes and E is the set of links. Let u and v be two adjacent nodes in G, then the Salton index between u and v is defined as

$$SI_{uv} = \frac{|N_u \cap N_v|}{\sqrt{|N_u| * |N_v|}} \tag{3}$$

where N_t is the set of neighbors of t and $|x|$ denotes the number of elements in set x.

The Salton index indicates that the larger the index value between two nodes is, the more common neighbors they have, thereby the two nodes will have a larger possibility in the same community. It is worth noting that some similar indexes can also be used here, such as Jaccard index [60] and Sørensen index [61]. These indexes are all based on the number of common neighbors, yet with different normalization methods. For the network depicted in Fig. 2, node u has a maximum priority and node v has a maximum similarity with u. The weak clique determined by nodes u and v consists of nodes u, v, a, b, and c. It is not difficult to find that the weak clique determined by u and v is often a "kernel" of a community, where the connections are much denser than other regions of communities. This means that all nodes of the weak clique determined by u and v have a larger possibility in the same community, even in a network with an unclear community structure.

In summary, in the proposed W-CPM the identification of weak cliques consists of two steps: 1) finding the node with a maximal priority and 2) selecting the one with a maximum similarity with the node. Once the two nodes are determined, a weak clique associated with the two nodes can be obtained according to Definition 1. The W-CPM starts to find another weak clique by repeating the above two steps in the remaining nodes until all nodes in the network are considered. Algorithm 2 presents the detailed procedure of identifying weak cliques in W-CPM.

B. Merging Weak Cliques Into Communities

Once all weak cliques in a network are identified, the proposed W-CPM starts to detect communities based on the found weak cliques. Due to the fact that the W-CPM only finds weak cliques instead of cliques in CPM, the W-CPM has to check whether the weak cliques can be merged into larger communities. To this end, a similarity metric between weak cliques is defined as follows.

Definition 4 (Weak Clique Similarity): Given an undirected network $G = (V, E)$, where V is the set of nodes and E is the set of links. Let C_1 and C_2 be two weak cliques in G, then the weak clique similarity between C_1 and C_2 is defined as

$$WS_{C_1C_2} = \frac{|V(C_1) \cap V(C_2)| + |E(C_1, C_2)|}{\min(|V(C_1)|, |V(C_2)|)} \tag{4}$$

where $V(C)$ is the set of nodes in weak clique C, $E(C_1, C_2)$ is the set of links between weak cliques C_1 and C_2, and $|x|$ indicates the number of elements in set x.

From the definition, we can find that the weak clique similarity not only considers the number of common nodes of weak cliques, but also takes the number of connections

Algorithm 2 $IdentifyWeakClique(G)$

```
Input: Network $G(V,E)$
Output: $WClique = \{W_i | W_i \subseteq V$ and $W_i$ is a weak clique$\}$
$WClique \leftarrow \emptyset$
for each $i \in V$ do
    $P(i) \leftarrow$ Compute the priority of node $i$ by Equation 2
end for
while $V != \emptyset$ do
    $u \leftarrow \arg\max_i(P(i))$
    $SI \leftarrow$Calculate Salton index for each neighbor $v$ of node $u$ by Equation 3
    $W_u \leftarrow \emptyset$
    while $SI\ != \emptyset$ do
        /*Select the node $v$ which has the maximal similarity with $u$ in set $SI$ */
        $v \leftarrow \arg\max_i(SI(i))$
        $W_u \leftarrow W_u \cup (\{u,v\} \cup N(u) \cap N(v))$
        Remove $W_u$ from $SI$
    end while
    $WClique \leftarrow WClique \cup W_u$
    Remove $W_u$ from $V$
end while
return $WClique$
```

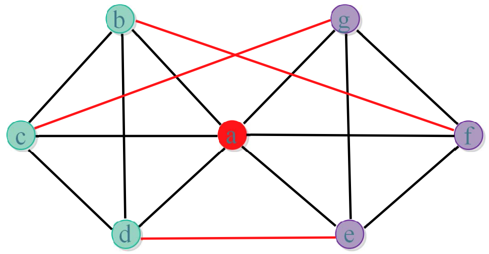


Fig. 3. Illustrative example where the proposed weak clique similarity may be advantageous over the strategy of merging cliques adopted in existing clique percolation algorithms. The network has two 4-cliques, one consisting of nodes a, b, c, d and the other consisting of nodes a, e, f, g. If the strategy of merging cliques adopted in existing clique percolation algorithms is used, the two 4-cliques will be considered as two communities. They will be merged into a community if the proposed weak cliques similarity is used in case the threshold $T < 1$ is set.

Algorithm 3 $Merge(wclique, S, T)$

```
Input: weak clique $wclique$, set of communities $S$, threshold $T$
/*Define two variables: $Comm$ is a community, $Container$ is a weak clique set*/
$Comm \leftarrow \emptyset$
$Container \leftarrow \emptyset$
$Comm \leftarrow Comm \cup wclique$
add $wclique$ into $Container$
while $Container\ != \emptyset$ do
    $temp \leftarrow$ select a weak clique from $Container$
    for each $neighbor\_temp \in N(temp)$ do
        if $WS_{temp,neighbor\_temp} > T$ and $neighbor\_temp$ unvisited then
            $Comm \leftarrow Comm \cup neighbor\_temp$
            add $neighbor\_temp$ into $Container$
            mark $neighbor\_temp$ as visited
        end if
    end for
    remove $temp$ from $Container$
end while
$S \leftarrow S \cup Comm$
```

between weak cliques into account. We propose this definition to address some potential weaknesses of the strategy for merging cliques adopted in the known algorithms based on clique percolation, where only the number of common nodes of cliques is considered. Fig. 3 illustrates a situation, where if the strategy in the known clique percolation algorithms is used, the clique consisting of a, b, c, d and the clique consisting of a, e, f, g will be considered as two communities. However, it is not difficult to find that these two cliques can be merged into a community since the links between them are so dense. The proposed weak clique similarity can correctly achieve the merging of the two cliques, if a threshold $T < 1$ is set. Note that we have $0 \leq WS_{C_1C_2} \leq \max(|V(C_1)|, |V(C_2)|)$ for weak cliques C_1 and C_2. In the experiment, we set the same threshold for all weak cliques in a network. Two weak cliques are merged into a community if the similarity between them is larger than T; otherwise, they are considered as in two communities. An empirical analysis of T on real-world networks is presented in Section IV.

With the similarity measure between weak cliques, it is easy to check whether the found weak cliques can be merged into a community by calculating the similarity between each pair of weak cliques. There is a little trick in calculating the similarity between weak cliques. If we have determined that weak cliques C_1 and C_2 can be merged into a community, and C_2 and C_3 can be merged into a community by calculating their similarities, then we do not need to calculate the similarity between C_1 and C_3 since these three weak cliques will be definitely merged into a community. This trick can considerably reduce the number of similarity calculations between weak cliques, thereby improving the computational efficiency of W-CPM. Algorithm 3 presents the detailed procedure of merging weak cliques into communities in W-CPM.

C. Complexity Analysis

The proposed W-CPM consists of the following two steps: 1) identifying the weak cliques in the network and 2) merging the weak cliques into communities. Let us assume that the network to be detected has n nodes, m links, the average degree d, the fraction of overlapping nodes o_n, and the number of communities each overlapping node belongs to o_m. The first step of W-CPM holds a time complexity of $O(dm)$, since both the calculation of the priority of each node and the identification of weak cliques based on the priority need a time complexity of $O(dm)$. In the second step, the weak cliques can be divided into o_n groups and we only need to check each pair of weak cliques in the same group, since only the weak cliques with at least one overlapping node can possibly

TABLE I

COMPARISON OF TIME COMPLEXITY BETWEEN SEVERAL EXISTING METHODS WITH W-CPM

Algorithm	Time Complexity
CPM [23]	$O(3.14^{n/3})$
SCP [43]	$O(3.14^{n/3})$
Reid's algorithm [44]	$O(3.14^{n/3})$
LC [39]	$O(d_{max}^2 n)$
GCE [28]	$O(mh)$
LFM [27]	$O(n^2)$
COPRA [34]	$O(vmlog(vm/n))$
SLPA [35]	$O(tm)$
FOCS [29]	$O(n+m)$
W-CPM	$O(dm)$

m=Number of links in the network,
n=Number of nodes in the network,
d=Averge degree of the network,
t=Predefined maximum number of iterations,
d_{max}=Maximum degree of the network,
h=Number of cliques,
v=Maximum number of communities a node can participate in.
Note that CPM, SCP and Reid's algorithm all need to identify the cliques in the network, whose computational cost is closely related to the average degree of the network. According to a result of Moon-Moser graphs with n node [42], Bron and Kerbosch showed that the total computing time of identifying cliques is proportional to $(3.14)^{n/3}$.

be merged into a community. Each group contains at most o_m weak cliques due to the fact that each overlapping node belongs to o_m communities. So, a total of $o_n o_m$ comparisons between weak cliques need to be performed. For comparing two weak cliques, the weak clique similarity between them needs to be calculated, which requires a time complexity of $O(d\mathtt{log}d)$ due to the fact that the weak cliques have an average size of d. This means that the second step of W-CPM holds a time complexity of $O(o_n o_m d\mathtt{log}d)$. Since $o_n = kn$, $0 < k < 1$, we often have $o_n o_m d\mathtt{log}d < dm$ for networks whose overlapping density and diversity are not very high. Therefore, the time complexity of the whole algorithm W-CPM is $O(dm)$. Table I shows a comparison of the time complexity between several existing methods and W-CPM.

IV. EXPERIMENTAL RESULTS

In this section, we verify the performance of the proposed W-CPM by comparing it with six state-of-the-art overlapping community detection algorithms, namely, SCP [43], Reid's algorithm [44], LC [39], SLPA [35], LFM [27], and FOCS [29]. Among the six compared algorithms, SCP and Reid's algorithm are two improved versions of the well-known CPM based on clique percolation theory, LC is a popular method based on link community, SLPA is a representative method of using the idea of label propagation, and LFM and FOCS are two local expansion and optimization-based methods for overlapping community detection.

A. Experimental Setting

The experiments are conducted on both synthetic and real-world networks with the size of nodes varying from thousands to hundreds of thousands. For each kind of networks, we compare both quality of found communities in terms of widely used performance indicators and computational efficiency with respect to runtime. Two popular performance indicators developed for evaluating the quality of obtained overlapping communities are adopted in the experiments depending on whether the true community structure is known. For all synthetic networks and some real-world networks whose true community structures are known, the extended normalized mutual information (NMI) suggested in [27] is used, while the overlapping modularity Q_{ov} developed in [34] is used for the other real-world networks since we do not know their true community structures. The larger the value of NMI and Q_{ov}, the better the quality of found communities.

For fair comparisons, the original implementations have been used for each of the compared algorithms. Further, all parameters of the six compared algorithms are set to the recommended values. Specifically, the maximum size of cliques in SCP and Reid's algorithm is set to $k = 4$ as recommended in [43] and [44]; the threshold of similarity between links in LC is set to 0.2 as recommended in [39]; the maximum number of iterations and the threshold r in SLPA are set to 100, 0.05, which were recommended in [35]; the parameter α in LFM for controlling the size of communities is set to $\alpha = 1.0$ as recommended in [27]; the minimum degree K for a node to build an initial community and the maximum overlap OVL allowed between communities in FOCS are set to the recommended values $K = 2$, $OVL = 0.6$ [29]; the threshold T of similarity between weak cliques is set to $T = 0.6$ in the proposed W-CPM unless otherwise specified. Note that all the experimental results reported in this paper are conducted on a PC with a 3.4 GHz Intel Core i3-3240 CPU 3.40 GHz, 4G internal storage and the Windows 7 SP1 32 bit operating system. The source code of the proposed W-CPM is available from the authors upon a request.

B. Performance on Synthetic Networks

The synthetic networks, we consider, are the LFR benchmark networks, which were proposed in [62] for testing the performance of algorithms to detect overlapping communities. In the following experiments, the computational efficiency of W-CPM is tested on two sets of LFR networks. These two sets of LFR networks are used to test the influence of network size and average degree on the efficiency of W-CPM, respectively. The first set of LFR networks consists of the networks whose size n varies from 10 000 to 100 000 with an interval 10 000 and the remaining parameters are fixed and set as follows. The average degree $d = 20$, the maximum degree $d_{\max} = 100$, the maximum community size $c_{\max} = 100$, the minimum community size $c_{\min} = 20$, the mixing parameter $\mu = 0.1$, the fraction of overlapping nodes $o_n = 0.1 \times n$, the number of communities each overlapping node belongs to $o_m = 2$, the average clustering coefficient $c = 0.7$, and the exponents of the power-law distribution of node degrees τ_1 and community sizes τ_2 are 2 and 1, respectively. The second set of LFR networks consists of the networks whose average degree d varies from 10 to 100 with an interval 10, where the network size n is set to 10 000, the minimum community size

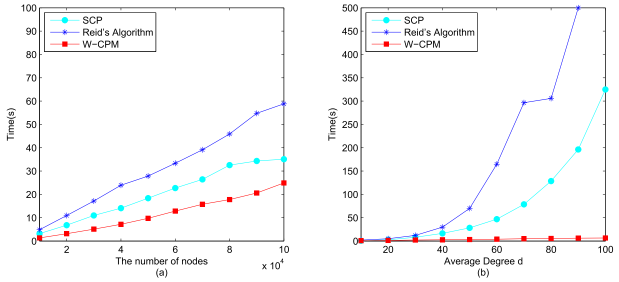


Fig. 4. Runtime(s) of the W-CPM method and two improved versions of CPM. (a) Runtime(s) on large-scale synthetic networks with different sizes. For all considered networks belonging to this set, the average degree d is set to $d = 20$. (b) Runtime(s) on large-scale synthetic networks with different average degrees. For all considered networks belonging to this set, the network size n is set to $n = 10\,000$.

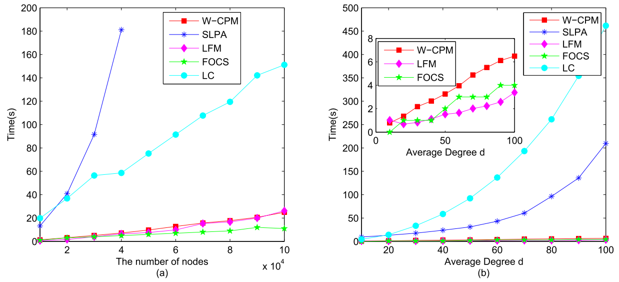


Fig. 5. Runtime(s) of the W-CPM method and four popular overlapping community detection algorithms that are not based on clique percolation theory. (a) Runtime(s) on large-scale synthetic networks with different sizes. (b) Runtime(s) on large-scale synthetic networks with different average degrees. These synthetic networks have the same setting as those in Fig. 4.

$c_{\min}$ is set to d, and the other parameters are set to the same values as in the first set of LFR networks.

Fig. 4 plots the runtime(s) of the W-CPM method and two improved versions of CPM, namely SCP and Reid's algorithm. As can be seen from the figure, the proposed W-CPM is expected to be less time consuming than the other two clique percolation-based algorithms, SCP and Reid's algorithm, especially when the average degree increases. The runtime of W-CPM almost keeps unchanged as the average degree increases, however, the runtime of SCP and Reid's algorithm will significantly increase. The main reason is attributed to the fact that W-CPM only needs to identify the weak cliques in a network, whereas SCP and Reid's algorithm have to find the cliques, which is very time consuming.

Fig. 5 presents the runtime(s) of the proposed W-CPM algorithm and four popular overlapping community detection algorithms that are not based on clique percolation theory, namely LC, SLPA, LFM, and FOCS. From the figure, we can find that the W-CPM takes less runtime than LC and SLPA, and also holds a competitive efficiency compared with state-of-the-art fast overlapping community detection algorithms, LFM and FOCS. The results in Fig. 5 also indicate that the computational cost of LFM, FOCS, and W-CPM is almost independent of the average degree of a network, which is promising for intensive networks. From Figs. 4 and 5, we can empirically confirm that the proposed W-CPM method is suited to coping with large-scale networks in terms of computational efficiency, especially for networks that are of both large scales and intensiveness.

Next, we compare the quality of found communities of the six overlapping community detection algorithms in terms of NMI. To this end, another three sets of LFR networks are generated by tuning the mixing parameter μ, the fraction of overlapping nodes o_n, and the number of communities each node belongs to o_m, respectively. These three sets of LFR networks are used to test the influence of community structure, overlapping density, and overlapping diversity on the performance of the W-CPM. For the mixing parameter μ, the larger the value of μ, the more unclear the community structure of LFR networks. An LFR network is considered to hold a clear community structure when $\mu \leq 0.5$, and an unclear community structure in case $\mu > 0.5$. We need to stress that the unclear community structure does not mean that the community structure can not be defined for LFR networks. For given network size and maximum community size, we can generate LFR networks with well-defined community structure for all values of mixing parameter with $\mu < 0.9$ [63]. To further verify the performance of W-CPM on different community sizes and network sizes, each set of LFR networks consists of four groups of networks, i.e., smaller network size and smaller community size ($n = 10\,000$, $c_{\min} = 10$, $c_{\max} = 50$), smaller network size and larger community size ($n = 10\,000$, $c_{\min} = 20$, $c_{\max} = 100$), larger network size and smaller community size ($n = 50\,000$, $c_{\min} = 10$, $c_{\max} = 50$), and larger network size and larger community size ($n = 50\,000$, $c_{\min} = 20$, $c_{\max} = 100$). The remaining parameters are fixed and set as follows. The average degree $d = 10$, the maximum degree $d_{\max} = 50$, the average clustering coefficient $c = 0.7$, and the exponents of the power-law distribution of node degrees τ_1 and community sizes τ_2 are 2 and 1, respectively. The parameters adopted in the above three sets of LFR networks are set according to those recommended in [19]. Note that we only present the quality of communities found by SCP in the following experiments, since for a given network SCP and Reid's algorithm always detect an identical result and the only difference between them is their computational efficiency.

Fig. 6 presents the NMI of the six algorithms on synthetic networks whose mixing parameter μ ranges from 0.1 to 0.8 with an interval 0.05, where o_n and o_m are set to $o_n = 0.1 \times n$ and $o_m = 2$. From this figure, we can find that the proposed W-CPM outperforms the existing clique percolation algorithms, SCP and Reid's algorithm, in terms of NMI accuracy. This result shows that weak cliques may be more suited than cliques for forming communities. The NMI accuracy of the proposed W-CPM is also more competitive than other algorithms that are not based on clique percolation theory on both synthetic networks with clear community structure and those with unclear community structure. It seems that the competitiveness of the W-CPM decreases as the community size of a network increases, however, the proposed W-CPM algorithm will still perform the best on synthetic networks with larger community size in case the community structure is unclear. Moreover, the superiority of the proposed W-CPM

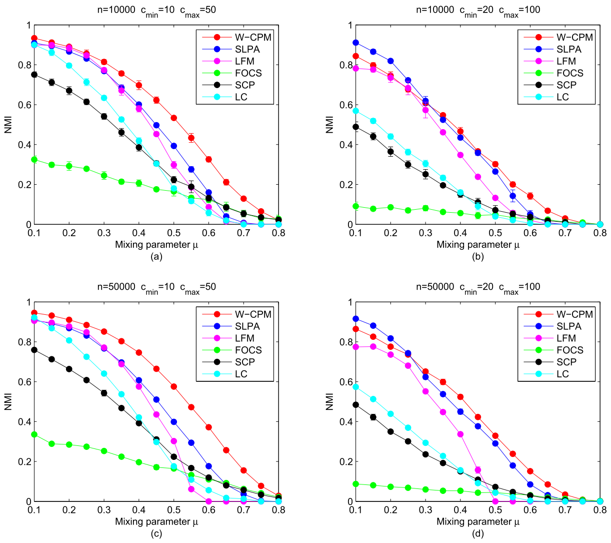


Fig. 6. NMI accuracy of the six algorithms on large-scale synthetic networks with different mixing parameters. Error bars show the standard deviations estimated from 20 networks. (a) NMI on networks with smaller size and smaller communities ($n = 10\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (b) NMI on networks with smaller size and larger communities ($n = 10\,000$, $c_{\min} = 20$, $c_{\max} = 100$). (c) NMI on networks with larger size and smaller communities ($n = 50\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (d) NMI on networks with larger size and larger communities ($n = 50\,000$, $c_{\min} = 20$, $c_{\max} = 100$).

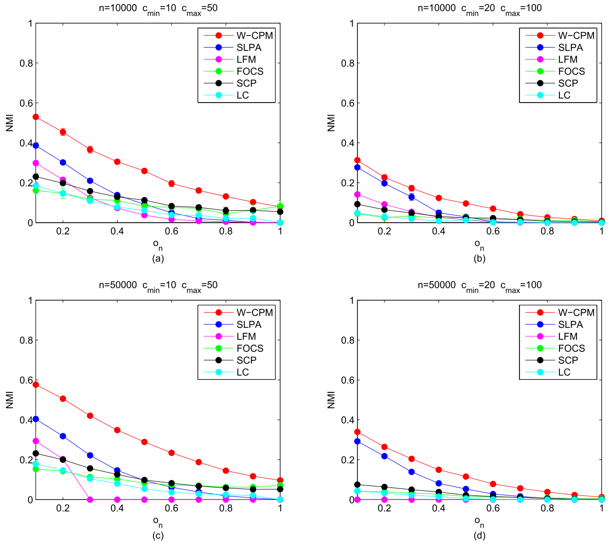


Fig. 7. NMI accuracy of the six algorithms on large-scale synthetic networks with different fractions of overlapping nodes. Error bars show the standard deviations estimated from 20 networks. (a) NMI on networks with smaller size and smaller communities ($n = 10\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (b) NMI on networks with smaller size and larger communities ($n = 10\,000$, $c_{\min} = 20$, $c_{\max} = 100$). (c) NMI on networks with larger size and smaller communities ($n = 50\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (d) NMI on networks with larger size and larger communities ($n = 50\,000$, $c_{\min} = 20$, $c_{\max} = 100$).

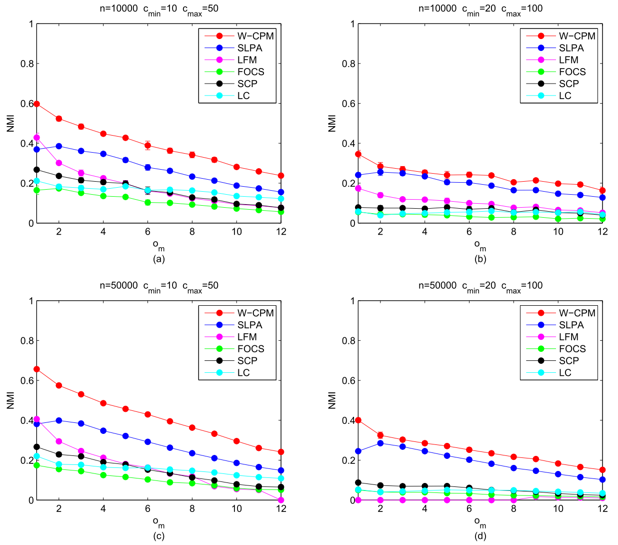


Fig. 8. NMI accuracy of the six algorithms on large-scale synthetic networks with different numbers of communities each overlapping node belongs to. Error bars show the standard deviations estimated from 20 networks. (a) NMI on networks with smaller size and smaller communities ($n = 10\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (b) NMI on networks with smaller size and larger communities ($n = 10\,000$, $c_{\min} = 20$, $c_{\max} = 100$). (c) NMI on networks with larger size and smaller communities ($n = 50\,000$, $c_{\min} = 10$, $c_{\max} = 50$). (d) NMI on networks with larger size and larger communities ($n = 50\,000$, $c_{\min} = 20$, $c_{\max} = 100$).

will increase as the network size becomes larger. This means that the proposed W-CPM is very promising, since most real-world networks are of large scales and hold an unclear community structure.

Fig. 7 plots the NMI accuracy of the six algorithms on synthetic networks whose fraction of overlapping nodes ranges from 0 to 1 with an interval 0.1, where the μ and o_m are set to $\mu = 0.5$ and $o_m = 2$. As shown in the figure, although the performance of all algorithms deteriorates as the overlapping density increases, the proposed W-CPM outperforms the compared five algorithms on all considered synthetic networks with different overlapping densities. These results illustrate the scalability of the W-CPM in the overlapping density.

Fig. 8 presents the NMI accuracy of the six algorithms on synthetic networks whose number of communities each overlapping node belongs to ranges from 1 to 12 with an interval 1, where the μ and o_n are set to $\mu = 0.5$ and $o_n = 0.1 \times n$. As can be seen from the figure, the proposed W-CPM outperforms the five compared algorithms on synthetic networks with larger number of communities each overlapping node belongs to, in case the networks hold an unclear community structure. The results shown in Fig. 8 confirm the competitiveness of the proposed W-CPM in detecting communities with higher overlapping diversity.

It is worth noting that all algorithms obtain a low NMI accuracy in Figs. 7 and 8, which is attributed to the fact that the networks considered in the two figures use a large mixing parameter value $\mu = 0.5$. As stated above, a large value of mixing parameter μ indicates an unclear community structure of the LFR networks, and thus the task of overlapping detection will become more difficult for overlapping detection algorithms. To show this fact, Fig. 9 presents the NMI values of the six compared algorithms on LFR networks with a small mixing parameter $\mu = 0.1$, where the rest parameters are set to the same values as those of LFR networks considered in Figs. 7 and 8. From the figure, it can clearly be found that all compared algorithms obtain a high NMI value on LFR networks with a small mixing parameter value.

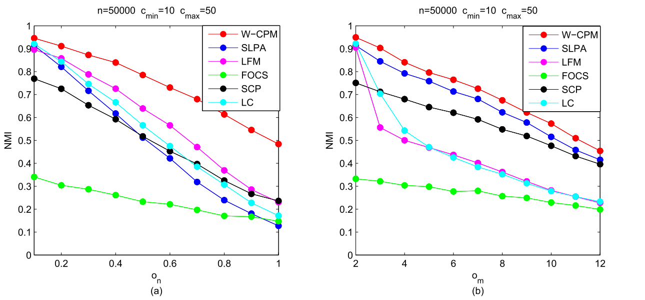


Fig. 9. NMI accuracy of the six algorithms on LFR networks with a small mixing parameter value $\mu = 0.1$. Error bars show the standard deviations estimated from 20 networks. (a) NMI on networks with different fractions of overlapping nodes. (b) NMI on networks with different numbers of communities each overlapping node belongs to.

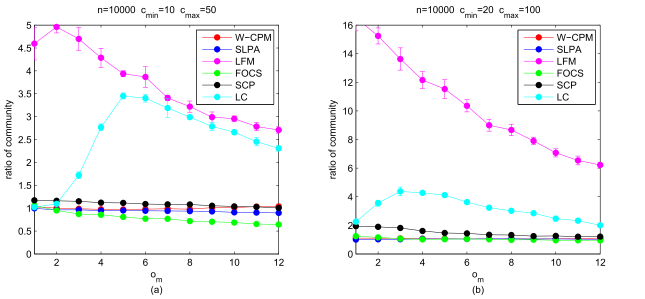


Fig. 10. Ratio of the detected to the known numbers of communities for large-scale synthetic networks with different numbers of communities each overlapping node belongs to. Error bars show the standard deviations estimated from 20 networks. Values over one are possible when more communities are detected than there are known to exist. (a) Ratio on networks with smaller communities ($n = 10\,000$, $\mu = 0.1$, $c_{\min} = 10$, $c_{\max} = 50$). (b) Ratio on networks with larger communities ($n = 10\,000$, $\mu = 0.1$, $c_{\min} = 20$, $c_{\max} = 100$).

Fig. 10 shows the ratio of the detected to the known numbers of communities of the six algorithms on synthetic networks with different number of communities each overlapping node belongs to. From the figure, we can find that just as done by SLPA, the W-CPM identifies almost the same number of communities as that of ground truth in the benchmark networks. The LFM, LC, and SCP easily produce more communities than the ground truth, while FOCS trends to generate less communities. This means that the number of communities found by W-CPM is closer to that of the ground truth.

The empirical results shown in Figs. 6–10 have demonstrated the competitive performance of the proposed W-CPM on large-scale LFR synthetic networks. In the following, we further verify the performance of W-CPM on small-scale LFR synthetic networks, which are original benchmark networks introduced in the LFR benchmark paper [62], [63]. Fig. 11 presents the NMI accuracy of the six algorithms on LFR synthetic networks with a size of 1000 and different community sizes, whose mixing parameter μ ranges from 0.1 to 0.8 with an interval 0.05, where o_n and o_m are set to $o_n = 0.1 \times n$ and $o_m = 2$ and the other parameters are set to the same values as the LFR networks used in Fig. 6. As can be seen from the figure, the proposed W-CPM also demonstrates a competitive performance on small-scale synthetic networks under both small and large community sizes, especially when the community structure is not clear. Similar to the case on large-scale synthetic networks, the superiority of W-CPM on small-scale networks will decrease as the community size increases, while it still performs better than the compared algorithms in case the networks do not have a clear community structure.

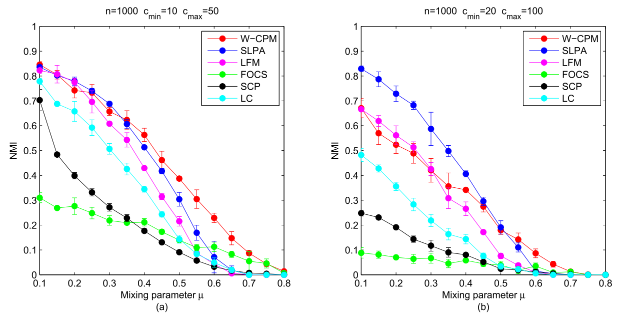


Fig. 11. NMI accuracy of the six algorithms on small-scale synthetic networks with different community sizes and mixing parameters. Error bars show the standard deviations estimated from 20 networks. (a) NMI on small-scale networks with smaller communities ($n = 1000$, $c_{\min} = 10$, $c_{\max} = 50$). (b) NMI on small-scale networks with larger communities ($n = 1000$, $c_{\min} = 20$, $c_{\max} = 100$).

From the above analysis, we can conclude that the proposed W-CPM algorithm is a promising overlapping community detection algorithm on large-scale synthetic networks, in terms of both computational efficiency and quality of found communities.

C. Performance on Real-World Networks

In this section, we verify the performance of the proposed W-CPM on several real-world networks, since real networks may have some topological properties which cannot be characterized by synthetic networks. The real-world networks that we consider here can be divided into two sets according to whether the ground truth is known. The first set of real-world networks consists of three large-scale real networks, namely com-Amazon, com-DBLP, and com-Youtube adopted from [64] and [65], whose ground truths are all known. The second set of real-world networks consists of nine real networks without a ground truth, namely, three real networks, PGP, blogs2, and word-association used in [34], five real networks, ca-HepTh, ca-GrQc, anybeat, soc-Epinions1, and web-Stanford from [64], and one co-authorship network, co-authorship_large used in [23]. Table II presents the detailed information of these real-world networks, including the number of nodes, the number of links, the average degree, and the number of communities in ground truth. Note that Q_{ov} is used to evaluate the quality of found communities for all the real networks and NMI is also provided for the real networks with ground truth. For all these real-world networks, the threshold T of similarity between weak cliques in W-CPM is set to $T = 0.3$.

Tables III–V present the experimental results of the seven compared algorithms on the two sets of real-world networks. From these tables, the following two observations can be

TABLE II

INFORMATION OF THE TWELVE REAL-WORLD NETWORKS USED IN THE EXPERIMENTS

Ground-truth	Network	Number of nodes	Number of links	Average degree	Number of communities
known	com-Amazon	334863	925872	5.53	151037
	com-DBLP	317080	1049866	6.62	13477
	com-Youtube	1134890	2987624	5.27	8385
unknown	ca-GrQc	5241	14484	5.53	unknown
	word-association	7205	31784	8.82	unknown
	ca-HepTh	9875	25973	5.26	unknown
	PGP	10680	24316	4.55	unknown
	anybeat	12645	49132	7.77	unknown
	blogs2	30557	82301	5.39	unknown
	co-authorship_large	30561	125959	8.24	unknown
	soc-Epinions1	75879	508837	13.41	unknown
	web-Stanford	281903	1992636	14.14	unknown

TABLE III

RESULTS OF THE SEVEN ALGORITHMS ON THREE REAL-WORLD NETWORKS WITH GROUND TRUTH

Network		LC	Reid's Algorithm	SCP	LFM	FOCS	SLPA	W-CPM
com-Amazon	runtime	46.7s	55.2s	45.7s	23.1m	**10.3s**	-	106.9s
	NMI	0.2451	0.2368	0.2368	0.2074	0.2236	0.1208	**0.3097**
	Q_{ov}	0.5571	0.3981	0.3981	0.6422	0.2569	-	**0.7430**
	community number	34.5K	23.1K	23.1K	52.5K	21.1K	30.5K	11.0K
com-DBLP	runtime	128.7s	48.4s	78.4s	24.2m	**9.4s**	-	103.6s
	NMI	0.1797	0.2196	**0.2196**	0.1289	0.2145	0.1191	0.1760
	Q_{ov}	0.3068	0.2987	0.2987	**0.5108**	0.3005	-	0.4263
	community number	55.6K	47.3K	47.3K	64.5K	24.2K	22.2K	56.5K
com-Youtube	runtime	4.9h	1.2h	23.0m	10.6h	**145.0s**	-	45.9m
	NMI	0.0052	0.0413	0.0413	0.0212	0.0335	0.0025	**0.0457**
	Q_{ov}	0.0122	0.1925	0.1925	0.2956	0.0287	-	**0.5637**
	community number	0.2K	7.4K	7.4K	228.4K	7.3K	39.9K	26.8K

The results of SLPA in the table are directly obtained from [29], since the method terminated with error on the conducted PC due to the limitation of internal storage. h, m and s denote hour, minute and second, respectively. K denotes a thousand.

TABLE IV

RUNTIME(S) OF THE SEVEN ALGORITHMS ON NINE REAL-WORLD NETWORKS WITHOUT GROUND TRUTH

Network	Runtime(s)						
	LC	Reid's Algorithm	SCP	LFM	FOCS	SLPA	W-CPM
ca-GrQc	0.85	0.63	1.06	0.27	**0.03**	3.29	0.28
word-association	3.33	1.09	0.25	0.75	**0.08**	7.37	0.41
ca-HepTh	2.57	1.22	**0.36**	0.95	0.50	9.80	0.58
PGP	2.38	1.35	0.66	1.20	0.60	9.64	**0.53**
anybeat	83.50	10.74	3.22	8.79	**0.63**	9.50	1.20
blogs2	5.11	3.88	**0.75**	11.77	1.96	91.41	1.80
co-authorship_large	15.00	4.99	2.18	16.00	**0.44**	96.41	2.31
soc-Epinions1	1240.50	3268.89	70.24	102.84	**7.52**	-	13.44
web-Stanford	1090.52	1088.22	2187.50	$> 12h$	-	-	**577.02**

The blanks in the table denotes that the method terminated with error before 12 hours.

made. First, the FOCS performs the best on these real-world networks in terms of computational efficiency with the only exception of web-Stanford, on which FOCS terminates with error. Compared with the six compared algorithms, the proposed W-CPM also holds a competitive efficiency on all considered real-world networks, especially when the network holds a slightly large average degree. The fact that the W-CPM is more suited to handling networks with a large average degree has also been observed on synthetic networks as shown in Figs. 4 and 5.

Second, W-CPM performs much better than the six compared overlapping community detection algorithms on both sets of real-world networks in terms of quality of found communities. The empirical results on the three real-world networks with ground truth demonstrate that the superiority of W-CPM over the six compared algorithms is independent of the adopted performance metrics Q_{ov} and NMI, despite that they often have a little inconsistence on some networks. The competitive performance of W-CPM may partially confirm that most real-world networks whose true community structures are still unknown hold an unclear community structure, since the W-CPM can achieve a better performance on networks without a clear community structure as shown in synthetic networks. Therefore, we can conclude that the proposed W-CPM outperforms the state-of-the-art algorithms on large-scale real-world networks, in terms of both computational efficiency and quality of found communities.

To take a closer look at the quality of communities detected by W-CPM, Figs. 12 and 13 plot the communities associated with Newman and Stauffer detected by the W-CPM in co-authorship_large network [23], where Newman and Stauffer are two famous scientists in complex networks and physics. As can be seen from the figures, Newman cooperates much with scientists from Israel (people in the community with yellow color) and New York (those in the community with blue color), but the cooperation between those from Israel and

TABLE V

Q_{ov} OF THE SEVEN ALGORITHMS ON NINE REAL-WORLD NETWORKS WITHOUT GROUND TRUTH

Network	Q_{ov}					
	LC	SCP	LFM	FOCS	SLPA	W-CPM
ca-GrQc	0.5145	0.4426	0.6426	0.3930	**0.7797**	0.7359
word-association	0.0653	0.1360	0.2249	0.0954	0.3617	**0.4759**
ca-HepTh	0.3254	0.2501	0.4626	0.2465	0.5976	**0.6091**
PGP	0.4603	0.4079	0.6775	0.2655	**0.8005**	0.6736
anybeat	0.0277	0.4757	0.2308	0.0394	0.1924	**0.6768**
blogs2	0.2587	0.1551	0.3654	0.0955	0.4736	**0.5484**
co-authorship_large	0.2766	0.3471	0.4116	0.2628	0.001	**0.4732**
soc-Epinions1	0.0653	0.4723	0.1478	0.0498	-	**0.7452**
web-Stanford	0.2305	0.7123	-	-	-	**0.8540**

The blanks in the table denotes that the method gives no result due to a termination with error or a undesired runtime.

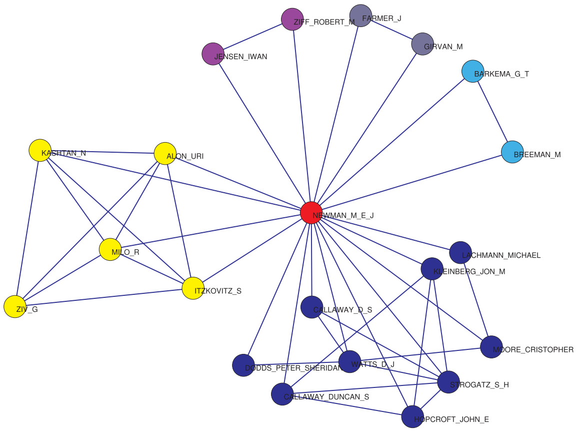


Fig. 12. Communities associated with Newman detected by W-CPM in co-authorship_large network. Five communities were found by W-CPM, and each color denotes a community.

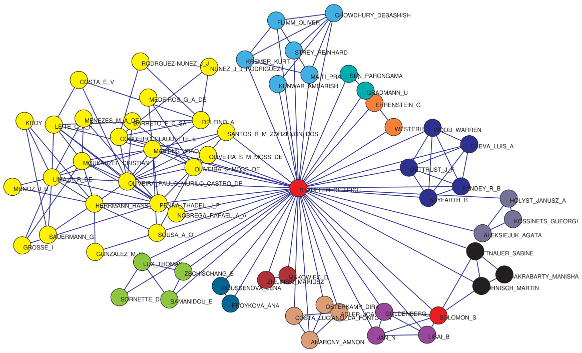


Fig. 13. Communities associated with Stauffer detected by W-CPM in co-authorship_large network. Twelve communities were found by W-CPM and each color denotes a community.

New York is very little, which may be partly due to the fact that the distance between New York and Israel is too far to do effective discussion. As for Stauffer, he has the coauthors from many different research fields of physics, including theoretical physics, statistical physics, and econophysics. So, we can conclude that Stauffer should have a wide range of research interests.

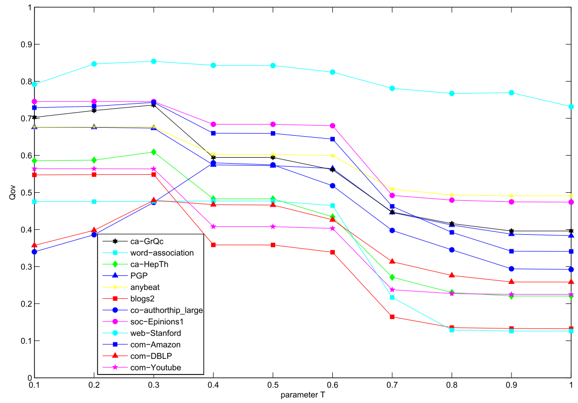


Fig. 14. Performance of W-CPM with different settings for threshold T of similarity between weak cliques on the 12 real-world networks.

D. Sensitivity of Parameter T in W-CPM

As mentioned above, we need to set a threshold T of similarity between weak cliques before the W-CPM is applied to a network. In the following, we empirically investigate the influence of T on the performance of W-CPM for real-world networks. Fig. 14 presents the performance of W-CPM with different settings for threshold T of similarity between weak cliques on the 12 real-world networks. As shown in the figure, we can find that T needs to be set to a small value in W-CPM on real-world networks. This is due to the fact that most real-world networks hold an unclear community structure, and thus, there are not a large number of links between weak cliques, which leads to a small similarity between weak cliques despite the fact that they can be merged into a community. From the empirical results on the 12 real-world networks, threshold T is suggested to be set to $T = 0.3$ on real-world networks.

V. CONCLUSION

In this paper, we have proposed a fast algorithm based on the idea of clique percolation for overlapping community detection in large-scale networks. In order to improve the computational efficiency in large-scale networks, the proposed algorithm finds weak cliques in the network instead of cliques and then uses these weak cliques to detect larger communities. A simple and efficient strategy has been suggested in the proposed algorithm to find the weak cliques in complex networks, enabling the proposed algorithm to be less time consuming than the known clique percolation algorithms, since the identification of cliques is NP-hard. For merging weak cliques into larger communities, a similarity between weak cliques has been suggested, by which the weak cliques belonging to the same community will be merged, and thus, communities hidden in complex networks could be correctly detected. Experimental results on both synthetic and real-world networks have demonstrated the superior performance of the proposed W-CPM over the competing approaches for detecting overlapping communities in large-scale networks in terms of both runtime and the quality of found communities.

The results of this paper suggest that the weak clique is an interesting idea for the existing clique percolation algorithms, since the proposed W-CPM algorithm based on weak cliques cannot only reduce the computational time, but also improve the quality of found communities. The performance of W-CPM is closely related to the strategy for merging weak cliques into larger communities. For this reason, a similarity between weak cliques has been suggested in W-CPM by considering both common nodes and the links between weak cliques. This similarity has shown the effectiveness in identifying which weak cliques belong to the same community and which ones do not belong to a community. However, it is still necessary to develop more effective strategies for merging weak cliques. There are also many other interesting problems related to the proposed W-CPM. For example, is it possible to use this method for different types of networks? What types of communities does W-CPM find? What is the shape of resulting communities of W-CPM? Can we use the definition of weak clique apart from community detection? It is also interesting to investigate extended versions of W-CPM by considering the W-CPM as a framework and testing things such as Jaccard index or modifications of centrality for priority/merging. All these questions and further more are now open, which need to be investigated further.

Acknowledgment

The authors would like to thank Dr. R. Cheng from the University of Birmingham, U.K., and Dr. C. Sun from the University of Surrey, U.K., for improving the writing of this paper.

References

[1] S. Harenberg *et al.*, "Community detection in large-scale networks: A survey and empirical evaluation," *Comput. Statist.*, vol. 6, no. 6, pp. 426–439, 2014.

[2] B. Yang, J. Liu, and D. Liu, "Characterizing and extracting multiplex patterns in complex networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 469–481, Apr. 2012.

[3] J. Qin, H. Gao, and W. X. Zheng, "Exponential synchronization of complex networks of linear systems and nonlinear oscillators: A unified analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 510–521, Mar. 2015.

[4] H. Liu and Y. Xia, "Optimal resource allocation in complex communication networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 7, pp. 706–710, Jul. 2015.

[5] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. La Porta, "Algorithms and applications for community detection in weighted networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 11, pp. 2916–2926, Nov. 2015.

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

[7] A. Prat-Pérez, D. Dominguez-Sal, J.-M. Brunat, and J.-L. Larriba-Pey, "Put three and three together: Triangle-driven community detection," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 3, p. 22, 2016.

[8] M. E. Newman, "Communities, modules and large-scale structure in networks," *Nature Phys.*, vol. 8, no. 1, pp. 25–31, 2012.

[9] B. Yang, J. Di, J. Liu, and D. Liu, "Hierarchical community detection with applications to real-world network analysis," *Data Knowl. Eng.*, vol. 83, pp. 20–38, Jan. 2013.

[10] D. Gómez, E. Zarrazola, J. Yáñez, and J. Montero, "A divide-and-link algorithm for hierarchical clustering in networks," *Inf. Sci.*, vol. 316, pp. 308–328, Sep. 2015.

[11] B. Yang, J. Liu, and J. Feng, "On the spectral characterization and scalable mining of network communities," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 2, pp. 326–337, Feb. 2012.

[12] P.-Y. Chen and A. O. Hero, "Phase transitions in spectral community detection," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4339–4347, Aug. 2015.

[13] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[14] D. Ji, Y. Sun, and D. Li, "An improved random walk based community detection algorithm," *Int. J. Multimedia Ubiquitous Eng.*, vol. 9, no. 5, pp. 131–142, May 2014.

[15] Y. Su, B. Wang, and X. Zhang, "A seed-expanding method based on random walks for community detection in networks with ambiguous community structures," *Sci. Rep.*, vol. 7, p. 41830, Feb. 2017.

[16] T. C. Havens, J. C. Bezdek, C. Leckie, K. Ramamohanarao, and M. Palaniswami, "A soft modularity function for detecting fuzzy communities in social networks," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1170–1175, Dec. 2013.

[17] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.

[18] Q. Cai, M. Gong, B. Shen, L. Ma, and L. Jiao, "Discrete particle swarm optimization for identifying community structures in signed social networks," *Neural Netw.*, vol. 58, pp. 4–13, Oct. 2014.

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

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

[21] A. H. Y. Tong *et al.*, "A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules," *Science*, vol. 295, no. 5553, pp. 321–324, 2002.

[22] C. Gaiteri *et al.*, "Identifying robust communities and multi-community nodes by combining top-down and bottom-up approaches to clustering," *Sci. Rep.*, vol. 5, pp. 1–8, Nov. 2015.

[23] 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, 2005.

[24] S. Zhang, X. Ning, and X.-S. Zhang, "Identification of functional modules in a PPI network by clique percolation clustering," *Comput. Biol. Chem.*, vol. 30, no. 6, pp. 445–451, 2006.

[25] J. Wang, B. Liu, M. Li, and Y. Pan, "Identifying protein complexes from interaction networks based on clique percolation and distance restriction," *BMC Genomics*, vol. 11, no. 2, p. S10, 2010.

[26] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding communities by clustering a graph into overlapping subgraphs," in *Proc. Int. Conf. Appl. Comput.*, 2005, pp. 97–104.

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

[28] C. Lee, F. Reid, A. McDaid, and N. Hurley. (2010). "Detecting highly overlapping community structure by greedy clique expansion." [Online]. Available: https://arxiv.org/abs/1002.1827

[29] S. Bandyopadhyay, G. Chowdhary, and D. Sengupta, "FOCS: Fast overlapped community search," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 2974–2985, Nov. 2015.

[30] L. Šubelj and M. Bajec, "Robust network community detection using balanced propagation," *Eur. Phys. J. B*, vol. 81, no. 3, pp. 353–362, 2011.

[31] M. He, M. Leng, F. Li, Y. Yao, and X. Chen, "A node importance based label propagation approach for community detection," in *Proc. 7th Int. Conf. Intell. Syst. Knowl. Eng.*, 2014, pp. 249–257.

[32] Z. Lin, X. Zheng, N. Xin, and D. Chen, "CK-LPA: Efficient community detection algorithm based on label propagation with community kernel," *Phys. A, Statist. Mech. Appl.*, vol. 416, pp. 386–399, Dec. 2014.

[33] S. Li, H. Lou, W. Jiang, and J. Tang, "Detecting community structure via synchronous label propagation," *Neurocomputing*, vol. 151, pp. 1063–1075, Mar. 2015.

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

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

[36] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," *Data Knowl. Eng.*, vol. 87, pp. 394–404, Sep. 2013.

[37] L. Yu, B. Wu, and B. Wang, "Topic model-based link community detection with adjustable range of overlapping," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2013, pp. 1437–1438.

[38] X. Zhang, N. Guan, W. Zhang, X. Huang, S. Wu, and Z. Luo, "Symmetric non-negative matrix factorization based link partition method for overlapping community detection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 2198–2203.

[39] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, Jun. 2010.

[40] T. S. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, p. 016105, Jul. 2009.

[41] D. Jin, B. Gabrys, and J. Dang, "Combined node and link partitions method for finding overlapping communities in complex networks," *Sci. Rep.*, vol. 5, Jan. 2015, Art. no. 8600.

[42] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *Commun. ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[43] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, "Sequential algorithm for fast clique percolation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 2, p. 026109, Aug. 2008.

[44] F. Reid, A. McDaid, and N. Hurley, "Percolation computation in complex networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 274–281.

[45] A. P. Francisco and A. L. Oliveira, "Fully generalized graph cores," in *Proc. 2nd Int. Workshop Complex Netw.*, 2011, pp. 22–34.

[46] K. Saito, T. Yamada, and K. Kazama, "Extracting communities from complex networks by the k-dense method," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 91, no. 11, pp. 3304–3311, 2008.

[47] J. Yang and J. Leskovec, "Community-affiliation graph model for overlapping network community detection," in *Proc. 12th IEEE Int. Conf. Data Mining*, Dec. 2012, pp. 1170–1175.

[48] Y. Zhang and D.-Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 606–614.

[49] H. Zhang, M. R. Lyu, and I. King, "Exploiting k-degree locality to improve overlapping community detection," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2394–2400.

[50] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2703–2716, Sep. 2017.

[51] E. Butun and M. Kaya, "A multi-objective genetic algorithm for community discovery," in *Proc. 7th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst.*, Sep. 2013, pp. 287–292.

[52] Y. Li, Y. Wang, J. Chen, L. Jiao, and R. Shang, "Overlapping community detection through an improved multi-objective quantum-behaved particle swarm optimization," *J. Heuristics*, vol. 21, no. 4, pp. 549–575, 2015.

[53] Y. Ju, S. Zhang, N. Ding, X. Zeng, and X. Zhang, "Complex network clustering by a multi-objective evolutionary algorithm based on decomposition and membrane structure," *Sci. Rep.*, vol. 6, Sep. 2016, Art. no. 33870.

[54] G. Chartrand and P. Zhang, *Chromatic Graph Theory*. Boca Raton, FL, USA: CRC Press, 2008.

[55] L. C. Freeman, "Centrality in social networks conceptual clarification," *Soc. Netw.*, vol. 1, no. 3, pp. 215–239, 1979.

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

[57] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Statist. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.

[58] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.

[59] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1986.

[60] P. Jaccard, "Etude de la distribution florale dans une portion des Alpes et du Jura," *Bull. del la Socit Vaudoise des Sci. Naturelles*, vol. 37, no. 1901, pp. 547–579, 1901.

[61] T. J. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons," *Biol. Skrifter*, vol. 5, pp. 1–34, Jan. 1948.

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

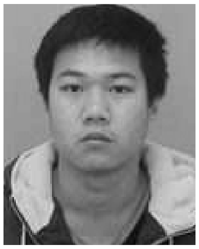
[63] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, p. 056117, Nov. 2009.

[64] J. Leskovec. (2010). *Stanford Network Analysis Project*. [Online]. Available: http://snap.stanford.edu

[65] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, 2015.

Xingyi Zhang received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

From 2013 to 2014, he was with the Nature Inspired Computing and Engineering Research Group, University of Surrey, Guildford, U.K. He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include unconventional models and algorithms of computation, multiobjective optimization, complex network, and membrane computing.



Congtao Wang received the B.Sc. degree from Anhui University, Hefei, China, in 2014, where he is currently pursuing the master's degree with the School of Computer Science and Technology. His current research interests include complex network and data mining.

Yansen Su received the B.S. degree in mathematics from Tangshan Teachers College, Hebei, China, in 2007, the M.S. degree in mathematics from the Shandong University of Science and Technology, Shandong, China, in 2010, and the Ph.D. degree in systems engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2014.

She is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei, China. Her current research interests include complex network, data mining, and systems biology.

Linqiang Pan received the Ph.D. degree from Nanjing University, Nanjing, China, in 2000.

Since 2004, he has been a Professor with the Huazhong University of Science and Technology, Wuhan, China. His current research interests include membrane computing, systems biology, and graph theory.

Hai-Feng Zhang received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2011.

He is currently a Professor with the School of Mathematics Science, Anhui University, Hefei. His current research interests include spreading dynamics, evolutionary game, and complex network.