A Local-Neighborhood Information Based Overlapping Community Detection Algorithm for Large-Scale Complex Networks

Fan Cheng, Congtao Wang, Xingyi Zhang, *Senior Member, IEEE*, and Yun Yang

***Abstract*— As the size of available networks is continuously increasing (even with millions of nodes), large-scale complex networks are receiving significant attention. While existing overlapping-community detection algorithms are quite effective in analyzing complex networks, most of these algorithms suffer from scalability issues when applied to large-scale complex networks, which can have more than 1,000,000 nodes. To address this problem, we propose an efficient local-expansion-based overlapping-community detection algorithm using local-neighborhood information (OCLN). During the iterative expansion process, only neighbors of nodes added in the last iteration (rather than all neighbors) are considered to determine whether they can join the community. This significantly reduces the computational cost and enhances the scalability for community detection in large-scale networks. A belonging coefficient is also proposed in OCLN to filter out incorrectly identified nodes. Theoretical analysis demonstrates that the computational complexity of the proposed OCLN is linear with respect to the size of the network to be detected. Experiments on large-scale LFR benchmark and real-world networks indicate the effectiveness of OCLN for overlapping-community detection in large-scale networks, in terms of both computational efficiency and detected-community quality.**

***Index Terms*— Overlapping-community detection, local neighborhood, large-scale network, linear complexity.**

I. Introduction

NETWORKS modeled as graphs with nodes (or vertices) and links provide a useful abstraction of the structure of various complex systems, ranging from social and computer networks to biological networks [1]–[4]. In the past decades, there has been increasing interest in the distinctive statistical properties that are common in many networks [5]–[7]. One such property is community structure (sometimes this property is also called cluster), and a number of studies have been conducted on community-structure detection, which has become important in network analysis [8]–[12].

In general, a given network is said to have community structure if there are groups of nodes (communities) that are more closely connected with each other than with the rest of the network [13], [14]. Community detection can be regarded as a network clustering problem, where each community corresponds to a cluster in the network [15]–[17]. Traditional community-detection techniques, such as hierarchical clustering [18], [19], random walks [20]–[22], and spectral clustering [23]–[25], partition a network so that each node belongs to exactly one cluster. However, it is well known that in several real-world applications, nodes may participate in multiple communities. Thus, there has been growing interest in overlapping-community detection, where nodes are allowed to belong to several different clusters [1], [26].

Existing methods for overlapping-community detection primarily include clique percolation [26]–[30], label propagation [31]–[33], and link partitioning [34]–[36]. Even though these algorithms have been proved effective, most of them suffer from scalability issues in large-scale complex networks owing to the high computational cost. In real-world scenarios, as the amount of obtained data increases, network size increases as well, and therefore overlapping-community detection becomes more challenging. Overlapping-community detection has several applications in large-scale networks [37], [38]. For example, in an online shopping system (such as Taobao), merchandise is represented by vertices in the network, a link between nodes indicates that merchandise is purchased by a customer, and a community represents a merchandise collection that belongs to the same purchasing pattern. Determining these merchandise collections is an overlapping-community detection problem in large-scale networks, as a merchandise item can belong to several different collections. In fact, the total number of such items in the shopping system is quite large and may even reach several millions.

Manuscript received March 27, 2019; revised January 17, 2020, May 7, 2020, and July 27, 2020; accepted November 13, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Elazouzi. Date of publication December 9, 2020; date of current version April 16, 2021. This work was supported in part by the National Key Research and Development Project under Grant 2018AAA0101302; in part by the National Natural Science Foundation of China under Grant 61822301, Grant 61672033, Grant 62076001, Grant 61976001, Grant 61876166, Grant 61663046 and Grant U1804262; and in part by the Humanities and Social Sciences Project of the Chinese Ministry of Education under Grant 18YJC870004. *(Corresponding author: Xingyi Zhang.)*

Fan Cheng, Congtao Wang, and Xingyi Zhang are with the Key Laboratory of Intelligent Computing Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: chengfan@mail.ustc.edu.cn; wctahuedu@163.com; xyzhanghust@gmail.com).

Yun Yang is with the National Pilot School of Software, Yunnan University, Kunming 650500, China (e-mail: yangyan19@hotmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNET.2020.3038756, provided by the authors.

Digital Object Identifier 10.1109/TNET.2020.3038756

As large-scale complex networks are quite widespread, efficient overlapping-community detection methods have been developed [27], [33], [35], [39]. Among them, approaches based on local expansion have been demonstrated to be highly suitable, as they identify communities only by using local topological structures [40]–[44]. This can significantly reduce computational time. Therefore, various promising local-expansion algorithms have been proposed for overlapping-community detection. Lancichinetti *et al.* [45] proposed a local-expansion overlapping-community detection algorithm, termed LFM, in which a community is expanded until a fitness function attains a local optimum. Whang *et al.* [46] developed an efficient overlapping-community detection algorithm by adopting new seeding strategies, and some other methods were presented in [47]–[49]. These algorithms have been demonstrated to be effective in large-scale networks, but their scalability in networks with more nodes should be further enhanced. To address this, Bandyopadhyay *et al.* [39] developed a fast overlapping-community search (FOCS) method with linear time complexity with respect to network size. Experiments demonstrated that FOCS was suitable for large-scale networks in terms of efficiency, even for networks with millions of nodes; however, the quality of the detected communities was often inferior to that of some existing methods based on local expansion.

In this paper, we propose a local-neighborhood information based expansion method, named OCLN. It allows the fast and high-quality detection of overlapping communities in large-scale networks. Specifically, the main contributions of this study are summarized as follows:

1) An expansion method based on local-neighborhood information is proposed for overlapping-community detection in large-scale networks. Unlike most existing local-expansion methods (as well as other methods, such as link community and label propagation), which use all neighbors to obtain a community, the proposed method only considers some key neighbors when expanding a given community. This implies that during the expansion, the proposed method evaluates significantly fewer nodes than existing methods; hence, it is considerably more efficient in large-scale networks.
2) Based on the aforementioned expansion method, an efficient local-expansion algorithm (OCLN) is developed for overlapping-community detection in large-scale complex networks. In OCLN, the proposed expansion method iteratively expands a community until no neighbor can be added. A measure for evaluating the probability of a node belonging to a community (the belonging coefficient) is also proposed for removing misidentified nodes during the expansions.
3) Theoretical analysis shows that the proposed algorithm has linear time complexity with respect to the number of links of a network. Experiments on large-scale synthetic and real-world networks demonstrate that the proposed OCNL is superior to existing overlapping-community detection algorithms in terms of both computational efficiency and community quality, particularly when the network has millions of nodes.

The remainder of the paper is organized as follows. In Section II, related work on existing overlapping-community detection algorithms is presented. Section III provides the details of the proposed algorithm. A comparison of OCLN with five state-of-the-art methods is carried out in Section IV. Section V concludes the paper.

II. Related Work

In the past decades, overlapping-community detection has attracted considerable attention owing to its wide application in social, computer, and biological networks [37], [50], [51]. The overlapping-community detection problem is formally defined as follows: Given an undirected, unweighted network (often represented as a graph)

$$G = (V, E) \tag{1}$$

where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, and $E = \{(i, j) \mid v_i \in V, v_j \in V, i \neq j\}$ is the set of links, the objective of overlapping-community detection is to obtain a high-quality community (group) partition $\Gamma = \{C_1, C_2, \ldots, C_k\}$ that maximizes EQ, which is a widely used metric for evaluating the quality of the detected overlapping community structure in complex networks [28], [52], [53]. A larger value of EQ implies higher quality. Specifically, EQ is formulated as

$$EQ = \frac{1}{2m} \sum_{i=1}^{k} \sum_{v \in C_i, u \in C_i} \frac{1}{O_v \cdot O_u} [A_{vu} - \frac{k_v \cdot k_u}{2m}] \tag{2}$$

where m is the number of links, A is the adjacent matrix of the network, O_v is the number of communities to which node v belongs, and k_v is the degree of node v. C_i denotes a community ($1 \leq i \leq k$, and k is the number of communities) that satisfies the following constraints:

$$C_i \subset V \; and \; C_i \neq \varnothing, i = 1, 2, \ldots k \tag{3}$$

$$\forall i \neq j \; and \; i, j \in \{1, 2, \ldots k\}, C_i \neq C_j \tag{4}$$

$$\bigcup_{i=1}^{k} C_i = V \tag{5}$$

$$\exists i \neq j \; and \; i, j \in \{1, 2, \ldots k\}, C_i \cap C_j \neq \emptyset \tag{6}$$

To solve this optimization problem, a large number of overlapping-community detection algorithms have been proposed based on different principles, such as clique percolation, label propagation, and link partitioning. Clique percolation is a deterministic community-detection method in which a k-clique community is defined as a set of nodes belonging to adjacent k-cliques. By allowing the nodes to have multiple community memberships, this method can be extended so that it can effectively solve the overlapping-community detection problem [26], [27]. The basic principle of label propagation is that the label of each node in a complex network is updated by replacing it by the label used by the greatest number of neighbors. In overlapping-community detection, a node can belong to v communities, where v is a predefined parameter [32], [33]. In link partitioning, links (instead of nodes) are partitioned into communities, where a node is overlapping if the links connected with it are assigned to more

than one cluster. Thereby, overlapping-community detection is converted into a disjoint clustering problem [35]. While these methods have been proved effective in detecting overlapping communities, most of them are difficult to be applied to real-world networks, since the real-world networks are often of large scales due to the available data becoming more and more.

To address this issue, Kumpula *et al.* [27] proposed a fast overlapping-community detection algorithm, termed SCP, by improving the popular clique percolation algorithm. The efficiency of SCP is achieved by decreasing the number of comparisons between cliques through a bipartite network when it is determined whether the obtained cliques can be merged into larger communities. Xie *et al.* [33] proposed an overlapping-community detection algorithm, termed SLPA, based on the label propagation algorithm LPA, where nodes quickly exchange labels according to a speaker–listener information propagation process. Ahn *et al.* [35] developed an efficient link partitioning algorithm, termed LC, whereby overlapping-community detection can be converted into non-overlapping-community detection. Experiments demonstrated that LC can be used to discover an overlapping community structure in large-scale complex networks.

The aforementioned overlapping-community detection algorithms have demonstrated their effectiveness in complex networks with more nodes. However, they are not as effective in excessively large networks, particularly in networks with millions of nodes. Recently, local-expansion methods have received considerable attention for overlapping-community detection in large-scale networks, as pointed out in [54], [55]. Unlike the aforementioned overlapping detection methods (such as clique percolation, label propagation, and link partitioning), which utilize the entire topological structure of a network, local-expansion methods detect communities by only considering local topological structures. The underlying principle of local-expansion methods is to start from a seed and, subsequently, iteratively expand it until a fitness function that measures the community quality reaches its optimum.

Using this principle, many local-expansion based overlapping-community detection algorithms have been proposed for large-scale networks. Lancichinetti [45] proposed a local-expansion algorithm, termed LFM, for overlapping-community detection. In LFM, a fitness function is defined based on the internal and external degrees of the nodes of a community, and a parameter α is adopted to control the size of the communities to be detected. Experiments demonstrated the effectiveness of LFM in large-scale networks [45]. Lee *et al.* [49] developed a greedy clique-expansion algorithm, termed GCE, where distinct cliques are identified as seeds, which are expanded by greedily optimizing a local fitness function. In addition, a distance between communities was also introduced so that seeds may not grow into similar communities, thus reducing the computational time [49]. Huang *et al.* [47] proposed a fast local-expansion algorithm for uncovering communities in large-scale networks, where the similarity of each pair of nodes is calculated only once by using a dynamical priority queue.

There are also some local-expansion methods reported to solve large-scale complex networks by adopting different seed strategies. Moradi *et al.* [48] proposed a local similarity score to determine seeds based on link prediction. Experiments on large-scale networks demonstrated that this algorithm could detect high-quality communities [48]. Whang *et al.* [46] developed a neighborhood-inflated seed-expansion algorithm, termed NISE, where two promising seeding strategies were proposed: graclus centers and spread hubs. In NISE, the personalized PageRank technique is adopted for fast community expansion. It was demonstrated that NISE outperformed several state-of-the-art algorithms on large-scale networks in terms of both computational efficiency and community quality [46].

The aforementioned local-expansion methods exhibit promising performance in detecting overlapping communities in large-scale networks. However, they also suffer from poor scalability in networks with several millions of nodes, as in most cases the local fitness function needs to be calculated repeatedly. To address this issue, Bandyopadhyay *et al.* [39] proposed a fast overlapping-community search algorithm, termed FOCS. In FOCS, the "leave" and the "expand" phase are iterated. At each iteration, multiple communities (rather than merely one community) are selected. This can considerably reduce the number of iterations and greatly expedite the computations [39]. This algorithm has a time complexity of $O(m)$, where m is the number of links in a network. Experiments indicated that FOCS was highly efficient in networks with millions of nodes. However, the quality of the detected communities still needs to be improved, particularly for excessively large networks, as will be shown in Table V in Section IV.

In this paper, we propose the OCLN overlapping-community detection algorithm (based on local-neighborhood information) for large-scale complex networks. To achieve fast community expansion, OCLN only uses some key neighbors in each expansion, instead of all nodes in the neighborhood. This resolves the problem of high expansion cost in most existing local-expansion methods for overlapping-community detection. For these algorithms, at each expansion all neighbors of the current community should be considered to determine whether they belong to the community. This is highly time consuming for a large-scale network, as the community will have a large number of neighbors. However, it can be observed that at each expansion some key neighbors of a community often have a high probability of belonging to the community, and the probability of the other neighbors is relatively low owing to the strength of their connection with the community.

Based on this observation, an efficient community expansion method is proposed for overlapping-community detection in large-scale networks. In this method, only neighbors of nodes added to the community in the current expansion are considered, instead of all neighbors of the community. The proposed method can considerably reduce the computational cost in each expansion, and thus it is suitable for large-scale networks. The computational efficiency of the proposed method on large-scale networks is discussed in Section IV.

III. Proposed Algorithm

The underlying principle of the proposed OCLN is to reduce the high expansion cost in existing local-expansion methods for large-scale networks. Hence, when the community is expanded, only some key neighbors of a community are considered instead of all neighbors. The proposed algorithm has a similar framework to that of most existing local-expansion methods. Specifically, it primarily consists of three phases: (1) initialization, (2) expansion, and (3) filtering. In the first phase, an initial community is constructed as a seed. Subsequently, in the second phase, the community is quickly expanded by using some key neighbors until no node in the network can be added to the community. In the final phase, nodes that are incorrectly assigned to the community are removed based on the proposed belonging coefficient measure. After this process is completed, a community is detected, and the three phases are iterated until all nodes in the network are assigned to a community. At that time, the proposed OCLN stops and converges to a stable state. Algorithm 1 presents the main procedure of the proposed OCLN.

Algorithm 1 Framework of OCLN Algorithm

Input:
Network $G(V, E)$
p: parameter used to control the overlapping community size.
α: parameter used to determine whether a node should be removed from expanding community.
Output: S $= \{S_i | S_i \subseteq$ V and S_i is a community $\}$
procedure OCLN(G, p, α)
  $S \leftarrow \emptyset$, $V_1 = V$
  CSET $\leftarrow \{i | (i, j \in V_1) \wedge (i \neq j) \wedge (\forall j : degree(j) \leq degree(i))\}$
  while do$core \in$ CSET, **and** core is unvisited
    $initialC \leftarrow$InitializeCommunity($core$)
    $expandC \leftarrow$ExpandCommunity($initialC, core, p$)
    $singleC \leftarrow$FilterCommunity($expandC, \alpha$)
    $S \leftarrow S \cup singleC$
    $V_1 = V - singleC$
    CSET $\leftarrow \{i | (i, j \in V_1) \wedge (i \neq j) \wedge (\forall j : degree(j) \leq degree(i))\}$
    mark each node in $singleC$ as visited
  end while
  return S
end procedure

In the following, we present the details of the three phases of the proposed OCLN algorithm.

A. Initializing Community Phase

In this phase, a set of nodes that are densely connected in the network is selected as a seed to be expanded. There are several community initialization strategies for local-expansion algorithms, such as maximal clique [49], graph center [46], and rank removal [55]. However, these strategies are not efficient for large-scale networks, particularly for networks with millions of nodes, owing to the high computational cost. In this paper, a fast strategy is proposed for community initialization in large-scale networks. Specifically, we first select a set consisting of a node with the maximal degree and all its neighbors. Compared with other criterions, such as betweenness [56] and k-shell [57], the degree is a simple yet effective metric whereby the core node can be quickly selected in a complex network. Subsequently, we remove the nodes that have fewer connections with nodes in the set (i.e., internal degree) than with the rest of the nodes in the network (i.e., external degree). By using the proposed strategy, we can obtain an initial community with densely connected nodes, and thus this initial community is usually the core of a network community. It is worth noting that the initial community obtained by the proposed strategy meets the definition of strong community [58].

Algorithm 2 Community Initialization

function InitializeCommunity($core$)
  $initialC = core \cup$ N($core$)
  for each $v \in initialC$ **do**
    if $k_{int}^v < k_{ext}^v$ **then**
      $initialC \leftarrow initialC - v$
    end if
  end for
  Return $initialC$
end function

Algorithm 2 presents the procedure of the community initialization strategy, where $core$ denotes the node with the maximal degree, $N(core)$ denotes the set of neighbors of $core$, and k_{int}^v and k_{ext}^v represent the internal and external degree of node v, respectively.

B. Expanding Community Phase

Once the initial community is identified, the proposed OCLN commences the fast expansion of the community using some key neighbors instead of all neighbors. In local-expansion algorithms, a community is iteratively expanded by adding new nodes from the neighbors of the current community. These added nodes are important in subsequent expansions, as the neighbors of these nodes have a higher probability of being added to the community compared to the other neighbors in the next expansion. The reason lies in the fact that in the former expansions, the neighbors of previously added nodes have already been checked and have not been added into the community. Therefore, compared with these neighbors, the neighbors of newly added nodes have a higher probability of being added to the community in the next expansion. In this study, the neighbors of newly added nodes are regarded as the key nodes and are termed nodes in the local neighborhood of the current community.

To determine which nodes in the local neighborhood can be added to the current community, a new metric termed local connectedness strength Δ is proposed, and all nodes in the local neighborhood satisfying $\Delta > 0$ join the current community. A larger value of Δ_v implies denser connections

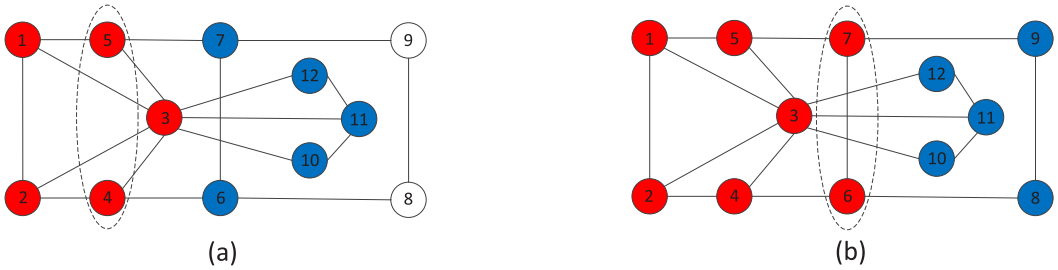

Fig. 1. Example to illustrate the efficiency of the proposed community-expansion method based on local-neighborhood information. It is assumed that a complex network has 12 nodes, and the current community consists of nodes 1, 2, 3, 4, and 5 (marked with red color). All its neighbors are composed of blue nodes 6, 7, 10, 11, and 12. Nodes 4 and 5 are the newly added nodes in the current community, and nodes 6 and 7 are the nodes in the local neighborhood of the current community. (a) In this expansion, only nodes 6 and 7 are considered in the proposed expansion method. (b) In the next expansion, only nodes 8 and 9 should be considered.

between node v in the local neighborhood and the newly added nodes in the current expansion. The definition of local connectedness strength is given as follows.

Definition 1 (Local Connectedness Strength): We assume that v is a node in the local neighborhood of the current community, d_{int}^{v} is the number of links between node v and newly added nodes $newAdd$, and d_{ext}^{v} represents the number of links between v and the rest of the nodes in the network. The local connectedness strength is defined as

$$\Delta_v = d_{int}^{v} - \frac{d_{ext}^{v}}{p}, \tag{7}$$

where p is a parameter for controlling the community size. A larger value of p implies that the detected community has more nodes. It should be stressed that to further reduce the computational cost, we only use the links with newly added nodes (instead of all nodes in the current community) to determine whether a node in the local neighborhood can be added to the community.

Using formula (7), all nodes in the local neighborhood satisfying $\Delta > 0$ are added to the current community, and this process is iterated until no node can be added. Algorithm 3 presents the details of the community expansion phase.

To illustrate the efficiency of the proposed community expansion method based on local neighborhood information, Fig. 1 shows an example network with 12 nodes, where the current community consists of nodes $\{1, 2, 3, 4, 5\}$, and nodes 4 and 5 are those added in the last expansion. As can be seen from Fig. 1(a), the current community $\{1, 2, 3, 4, 5\}$ has five neighbors: 6, 7, 10, 11, and 12; Hence, all five nodes should be considered to determine whether they can be added to the community in most existing local-expansion methods. By contrast, in the proposed method, only nodes 6 and 7 should be considered because nodes 10, 11, and 12 are not neighbors of the newly added nodes 4 and 5. According to the local connectedness strength, both nodes 6 and 7 are added to the current community because they have $\Delta > 0$, and the current community becomes $\{1, 2, 3, 4, 5, 6, 7\}$, as shown in Fig. 1(b). In the next expansion, the newly added nodes of community $\{1, 2, 3, 4, 5, 6, 7\}$ are 6 and 7. For community $\{1, 2, 3, 4, 5, 6, 7\}$, the nodes in its entire neighborhood and local neighborhood are $\{8, 9, 10, 11, 12\}$ and $\{8, 9\}$, respectively. This implies that the proposed expansion method only considers nodes $\{8, 9\}$, whereas nodes $\{8, 9, 10, 11, 12\}$ should be considered in most existing local-expansion methods.

Algorithm 3 Expanding Community

```
function ExpandCommunity(initialC, core, p)
    expandC ← ∅
    newAdd ← initialC − core
    while newAdd ≠ ∅ do
        candidate ← ∅
        for each v ∈ newAdd do
            for each v_k ∈ N(v) do
                candidate ← candidate ∪ {v_k}
            end for
        end for
        Add ← ∅
        for each v ∈ candidate do
            if d_int^v − d_ext^v / p > 0 and v ∉ expandC then
                Add ← Add ∪ {v}
            end if
        end for
        expandC ← expandC ∪ Add
        newAdd ← Add
    end while
    return expandC
end function
```

It follows that the proposed local-neighborhood information based community expansion method can significantly reduce the computational cost in each expansion; thus, it is suitable for overlapping-community detection in large-scale networks. It should be noted that the proposed fast expansion may result in nodes being incorrectly identified. Figure 2 shows such an example, where the current community consists of nodes $\{1, 2, 3, 4, 5\}$. It is seen that node 1 should not be added to the community $\{2, 3, 4, 5\}$, as node 1 is more densely connected with the set of nodes $\{6, 7, 8, 9, 10\}$. To address this issue, in the following we propose a filtering operation to remove incorrectly identified nodes.

C. Filtering Phase

After community expansion is completed, the proposed OCLN removes the nodes that are not well connected with the community. To determine the link strength of a node in

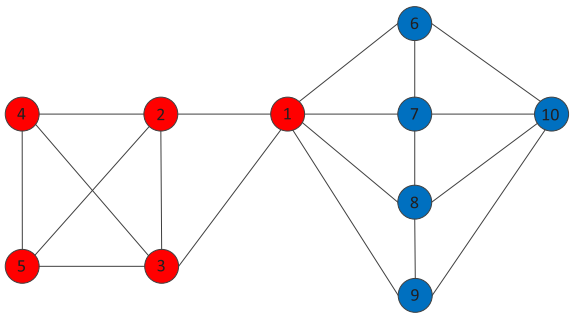

Fig. 2. Example of incorrectly identified nodes in the community after the expansion phase.

the community, we suggest the belonging coefficient, which is defined as follows:

Definition 2 (Belonging Coefficient): Given a community C, let v be a node of C, $N(x)$ be the set of neighbors of x, and k_{int}^{x} denote the number of links of x in C. Then, the belonging coefficient of v in C is defined as

$$\textit{belonging coefficient(v)} = \frac{\sum_{x\in(N(v)\cap C)} \frac{k_{int}^{x}}{|N(x)|}}{|N(v)|}, \qquad (8)$$

where $|X|$ denotes the number of elements in the set X.

According to the above definition, the belonging coefficient of a node can measure the probability of the node staying in a community. Only neighbors of the node are considered in the calculation of the belonging coefficient to reduce the computational cost for large-scale networks. A larger value of the belonging coefficient implies a higher probability that the node stays in the community. The definition of the belonging coefficient is motivated by an observation in many real-world networks. For example, in a social network, there is a person who has several friends. If most of his friends are highly active in the community, he has a high probability of joining the community. By contrast, if the majority of these friends do not have any interest in the group, this will obviously affect the person's decision, and he will probably not join the group. This observation inspires us to only use neighbors of a node to evaluate the possibility of the node staying in a community suggested in Definition 2.

Based on the belonging coefficient, incorrectly identified nodes can be filtered out from the community by setting a threshold α. For example, node 1, which is incorrectly assigned to community $\{1, 2, 3, 4, 5\}$ shown in Fig. 2, will be removed from the community when the threshold α is set to 0.5, as the belonging coefficient of this node equals to $\frac{2}{5}$, whereas the belonging coefficients of the other nodes in the community are all larger than 0.5. Algorithm 4 presents the procedure of the filtering phase of the proposed OCLN.

Algorithm 4 Filtering Community

function FILTERCOMMUNTIY($expandC, \alpha$)
  $singleC \leftarrow \emptyset$
  for each $v \in expandC$ **do**
    $neighborTemp \leftarrow \mathrm{N}(v) \cap expandC$
    $bcoefficient \leftarrow 0$
    for each $v_k \in neighborTemp$ **do**
      $bcoefficient \leftarrow bcoefficient + \frac{k_{int}^{v_k}}{|N(v_k)|}$
    end for
    $bcoefficient \leftarrow \frac{bcoefficient}{|N(v)|}$
    if $bcoefficient > \alpha$ **then**
      $singleC \leftarrow singleC \cup \{v\}$
    end if
  end for
  return $singleC$
end function

D. Theoretical Analysis of Proposed OCLN

In this subsection, we will provide a theoretical analysis of OCLN. It will be demonstrated that the proposed method can greatly reduce computational complexity without significantly affecting the accuracy of OCLN in community detection. Specifically, the theoretical analysis includes a theorem and a comment. Theorem 1 indicates that during each expansion, the community obtained by the proposed local-neighborhood method is the same as that obtained by considering all the neighbors. In Comment 1, we highlight the fact that by modifying the definitions of d_{int}^{v} and d_{ext}^{v}, the computational efficiency of the proposed method is further enhanced, but the quality of the obtained community slightly deteriorates.

Theorem 1: Let $G(V, E)$ be an undirected, unweighted network, and let $com_t^{LocalNeigh_t}$ and $com_t^{AllNeigh_t}$ be the community obtained by the proposed local-neighbor strategy and that considering all the neighbors after the t-th expansion, respectively. Then, for any $t \geq 1$, $com_t^{LocalNeigh_t}$ is same as $com_t^{AllNeigh_t}$.

Comment 1: It should be noted that in the proposed method, to improve the efficiency of OCLN, we provide new definitions of d_{int}^{v} and d_{ext}^{v}. Specifically, d_{int}^{v} denotes the number of links between node v and newly added nodes $newAdd$, and d_{ext}^{v} is the number of links between v and the rest of the nodes in the network. By only considering the degrees with respect to newly added nodes (not all the nodes in the current community), we can further reduce the computational cost of OCLN. However, these new definitions slightly degrade the quality of the obtained community.

The detailed proof of Theorem 1 is given in Supplementary A, and the empirical verification of Theorem 1 and Comment 1 on large-scale synthetic networks is provided in Supplementary B.

E. Complexity Analysis

Let $G(V, E)$ be a given undirected, unweighted network to be detected, where $n = |V|$ is the number of nodes, $m = |E|$ is the number of links, and the average degree of the network is k. The proposed method adopts an adjacency list to store the graph; this is a commonly used data structure in complex networks. Then, under this data structure, the time complexity of OCLN is as follows: In the first initializing community phase, we should evaluate n_c nodes, where n_c is the number of nodes in the initial community. For each node, we should calculate its internal and external degree, whose

TABLE I
COMPARISON OF TIME COMPLEXITY FOR VARIOUS EXISTING METHODS WITH OCLN

Algorithm	Time Complexity
SCP [27]	$O(3.14^{n/3})$
LC [35]	$O(k_{max}^2 n)$
GCE [49]	$O(mh)$
LFM [45]	$O(n^2)$
COPRA [32]	$O(vmlog(vm/n))$
SLPA [33]	$O(tm)$
FOCS [39]	$O(m)$
NISE [46]	$O(\sum_{i=1}^{j} max_{\varepsilon} links(C_i, V_C))$
OCLN	$O(m)$

TABLE II
FORMAL NOTATIONS USED IN TABLE I

Notation	Description
m	Number of links in the network
n	Number of nodes in the network
k	Average degree of the network
h	Number of cliques in the network
v	Maximum number of communities in which a node can participate
t	Predefined maximum number of iterations
k_{max}	Maximum degree of the network
j	Number of communities
ε	Number of links in community
$links(C_i, V_C)$	Number of links between community C_i and V_C

time complexity is $O(k)$. Thus, the total time complexity of the first phase is $O(k \times n_c)$. For the second (expanding community) phase, we only need to consider the nodes in $newAdd$ set, whose size is $O(n_c)$. For each node v in $newAdd$ set, we should calculate its local connectedness strength $\triangle_v$ to determine whether it can be expanded into current community. To this end, we need to calculate d_{int}^v and d_{ext}^v, whose time complexity is $O(k)$. Therefore, the computational cost of the second phase is also $O(n_c \times k)$. In the filtering phase, we should calculate the belonging coefficient of each node in the community after expansion, with a time complexity of $O(k \times n_c)$. Hence, the time complexity of detecting a community is $O(k \times n_c + k \times n_c + k \times n_c)$. As $O(k \times n_c) = O(m_c)$ (m_c is the number of links in the community), we have a time complexity of $O(m_c)$ for discovering a community in the proposed OCLN. The three phases above are repeated until all nodes in the network G are assigned to a community. Therefore, the proposed OCLN has a time complexity of $O(m)$ for detecting all communities in a network. Table I presents the time complexity of the proposed OCLN and several popular overlapping community detection algorithms. From the table we can find that the proposed method has the competitive time complexity among different algorithms. The reason is mainly attributed to the fact that in the proposed OCLN only some local neighbors instead of all the neighbors are evaluated to expand the community, which reduces the computational cost of OCLN greatly.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we will evaluate the performance of the proposed OCLN algorithm on large-scale complex networks by comparing it with five state-of-the-art overlapping-community detection algorithms. In the following, we first present the experimental setting, which includes comparison algorithms, evaluation metrics, parameter settings, and the test datasets. Then, we discuss the sensitivity of the parameters in the proposed algorithm. Finally, we provide the results by different algorithms on large-scale synthetic and real-world networks.

A. Experimental Setting

1) Comparison algorithms: Five popular algorithms are compared with the proposed OCLN in the following experiments. Similar to OCLN, all the five comparison algorithms used in this paper can be used to address the overlapping community detection problem defined in Section 2. The details of these algorithms are summarized as follows:

SCP [27] is an improved version of the well-known clique percolation algorithm CPM [26]. For a given network, SCP and CPM yield the same community-detection results. The main difference between them is their computational efficiency, as SCP adopts a bipartite network to reduce the number of comparisons between cliques.

LC [35] is an efficient link-partitioning method for overlapping-community detection. The LC algorithm has been demonstrated to be suitable for overlapping-community detection, particularly for networks with a high overlapping degree.

LFM [45] is a local expansion algorithm for overlapping-community detection. This method adopts a random node as the seed and expands a community from the seed until the fitness function arrives attains a local maximum.

NISE [46] is an overlapping-community detection algorithm based on local expansion, where two new seed strategies were developed to determine appropriate seeds. A personalized PageRank clustering method was also adopted in NISE to achieve fast community expansion. In NISE, the number of detected communities should be known in advance.

FOCS [39] is an overlapping-community search algorithm with linear time complexity with respect to the number of links. It has high computational efficiency in large-scale networks, even in networks with millions of nodes.

2) Evaluation metrics: We evaluate the performance of the compared algorithms on both large-scale synthetic and real-world networks. For each type of network, we report both the quality of the detected overlapping communities and the computational efficiency with respect to runtime.

As the ground truth of all considered networks is known, we adopt the normalized mutual information (NMI) [45], accuracy [59], and Jaccard index [60] to evaluate the quality of the detected communities. These are three commonly used metrics in overlapping-community detection. Larger values of these metrics imply better performance. If these metrics are equal to 1, then the community detection result of an algorithm is the same as the ground truth.

3) Parameter setting: For each comparison algorithm, one needs to set a few parameters. To be specific, the SCP requires to set the maximum size of cliques k. The LC algorithm requires the specification of the threshold of similarity

between links. LFM needs parameter α to be set, which is used to control the size of community. In FOCS, there are two parameters to be set: the minimum degree K for a node to construct an initial community, and the maximum allowed overlapping ratio OVL between communities. In NISE, we should specify the number of detected communities, and select the appropriate seeding strategy. In the proposed OCLN, there are two parameters: p for controlling the community size and the threshold α for removing incorrectly assigned nodes in the filtering phase.

To ensure fair comparisons in the experiments, for each baseline we perform community detection by testing a variety of parameter values (in the range suggested in the original studies), and select the value that results in the best performance. Specifically, the maximum size of cliques in SCP is set to $k = 4$, which is also the recommended value in [27], [29], the threshold of similarity between links in LC is set to 0.2, the parameter α in LFM is set to $\alpha = 1.0$, and the parameters K and OVL in FOCS are set to $K = 2$ and $OVL = 0.6$, respectively. For the NISE method, we use the "spread hubs" seeding strategy, and the number of communities for NISE is set to the true community number. For the proposed algorithm, p and α are set to $p = 4$, and $\alpha = 0.2$, respectively. All the experiments are conducted on a PC with a 3.4 GHz Intel Core i3-3240 CPU 3.40GHz, 4 GB internal storage, and the Windows 7 SP1 32 bit operating system. The source code of the proposed OCLN is available from the website https://github.com/BIMK/OCLN.

4) Test datasets: The algorithms under comparison are tested on both large-scale synthetic benchmark networks and real-world networks. The former are the LFR networks developed by Lancichinetti *et al.* in [61], which are the most widely adopted benchmark networks for testing the performance of overlapping-community detection algorithms. Compared with other synthetic networks, the LFR networks can reflect important features of complex real-world systems, as the distributions of the node degree and community size in these networks are both power laws with tunable exponents. Specifically, we use two groups of LFR networks to evaluate the performance of the proposed OCLN.

The first group of LFR networks is used to test the computational efficiency and quality of overlapping communities by different algorithms on large-scale complex networks. It contains three network sets: LFR-N, LFR-D, and LFR-μ. The LFR-N set consists of 12 LFR networks. Their node size n varies from 100000 (0.1 M) to 1200000 (1.2 M) with an interval of 100000. The other parameters of the LFR networks are fixed as follows: Mixing parameter $\mu = 0.1$, average degree $k = 10$, maximum degree $k_{\max} = 50$, minimum community size $\min_c = 20$, maximum community size $\max_c = 20$, fraction of overlapping nodes $o_n = 0.1 \times n$, number of communities to which each overlapping node belongs $o_m = 2$, and exponent of the power-law distribution of node degree $\tau_1 = 2$ and community size $\tau_2 = 1$. The LFR-D set consists of 15 LFR networks, where the average degree varies from 10 to 80 with an interval of 5. The other parameters are fixed as follows: $n = 100000, \mu = 0.1, k_{max} = 200, min_c = k, max_c = 200, o_n = 0.1 \times n, o_m = 2, \tau_1 = 2$, and $\tau_2 = 1$. The LFR-μ set consists of nine LFR networks, where the mixing parameter μ ranges from 0.1 to 0.5 with an interval 0.05. The other parameters are fixed as follows: $n = 1200000$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$).

The second group is used to verify the influence of overlapping diversity on the performance of the six algorithms under comparison. Accordingly, four LFR networks sets (LFR-Om1, LFR-Om2, LFR-Om3, and LFR-Om4) are generated in this group. Each LFR set consists of seven LFR networks, where o_m ranges from 2 to 8 with an interval of 1. The only difference between the four LFR sets are the network size n and the mixing parameter μ. Specifically, these two parameters for the four LFR sets are set as follows: (1) LFR-Om1 ($n = 100000$, $\mu = 0.1$), (2) LFR-Om2 ($n = 100000$, $\mu = 0.3$), (3) LFR-Om3 ($n = 500000$, $\mu = 0.1$), and (4) LFR-Om4 ($n = 500000$, $\mu = 0.3$). The remaining parameters of the four network sets are set as follows: average degree $k = 10$, maximum degree $k_{\max} = 50$, minimum community size $\min_c = 20$, maximum community size $\max_c = 100$, fraction of overlapping nodes $o_n = 0.1 \times n$, and exponent of the power-law distribution of node degree $\tau_1 = 2$ and community size $\tau_2 = 1$.

The large-scale, real-world networks we adopt in the experiments are Amazon, DBLP, YouTube, LiveJournal, and Orkut, which are publicly available from the Stanford network dataset collection.[1] These networks are undirected and unweighted, and their ground truth is known. They are summarized as follows.

Amazon. This network is collected by crawling the Amazon website. If a product i is frequently co-purchased with product j, then the network contains an undirected link between i and j. Each ground-truth community can be defined to be a product category that Amazon provides.

DBLP. This is a scientific collaboration network generated from research papers in computer science. A link is constructed between author i and author j if they publish at least one joint paper. In DBLP, each publication venue (i.e., journal or conference) can be considered a ground-truth community.

YouTube. This is a social network obtained from a video-sharing website, where users form friendship with each other. Users can create groups that other users can join, and the ground-truth communities are defined as these user-defined groups.

LiveJournal. This is a free on-line blogging network where users declare friendship with each other. Ground-truth communities are groups explicitly created by users based on common interest topics, affiliations, and geographical regions.

Orkut. This is a free on-line social network where users form friendship with each other. Orkut also allows users to form a group that other members can join. Ground-truth communities are defined as these user-defined groups.

These networks have excessively large size, ranging from hundreds of thousands to millions of nodes. Related basic information is listed in Table III.

[1] http://snap.stanford.edu/data/index.html

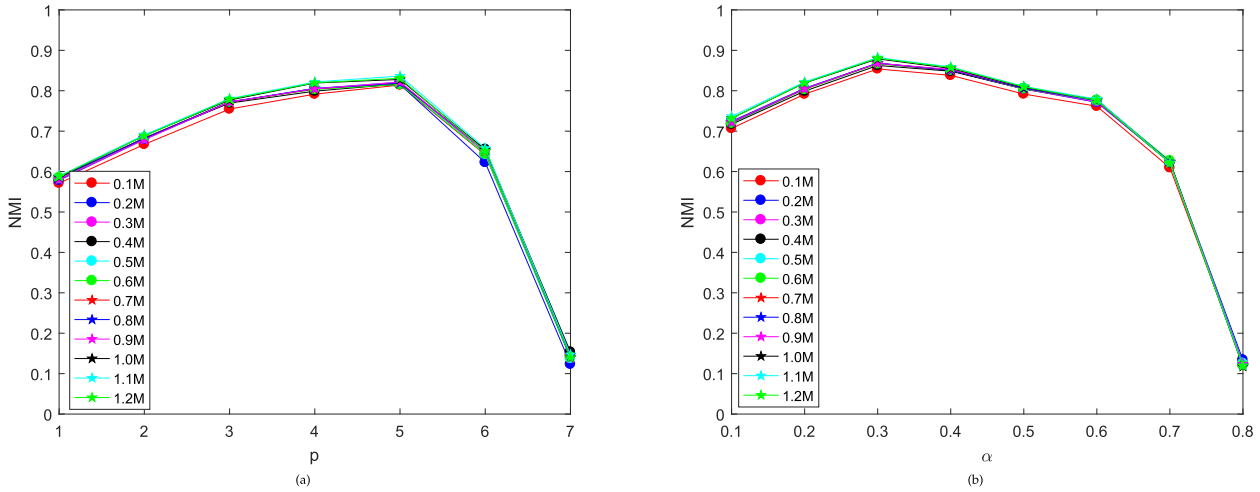

Fig. 3. NMI values of OCLN with different settings for parameters p and α on the large-scale LFR-N set, where n varies from 0.1M to 1.2M with an interval of 0.1M. Other parameters are fixed as follows: $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 20$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$. (a) NMI on the networks with different p. (b) NMI on the networks with different α.

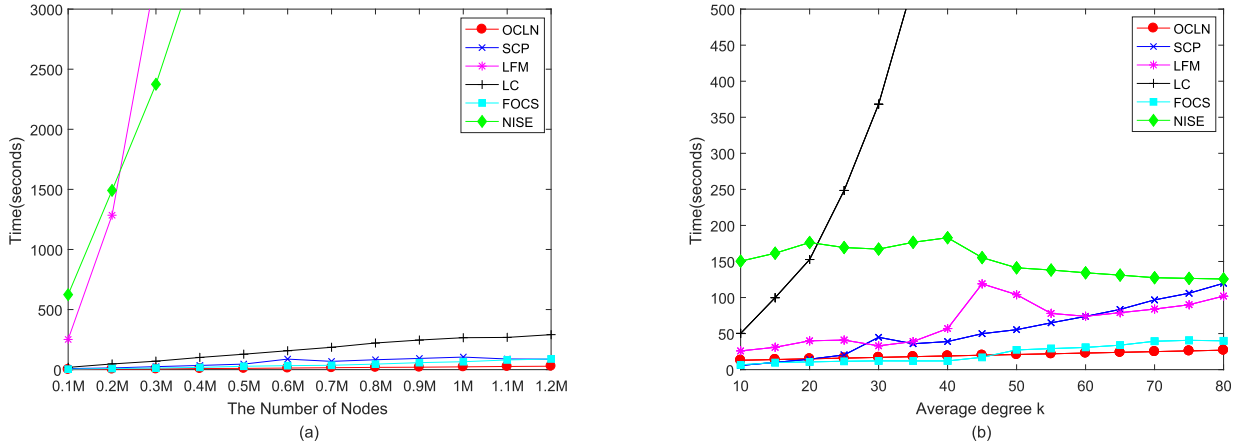

Fig. 4. Runtime (s) of the six algorithms on large-scale LFR-N and LFR-D sets. (a) Runtime on the LFR-N set, where n varies from 0.1M to 1.2M with interval 0.1M. Other parameters are fixed as follows: $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 20$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$. (b) Runtime on the LFR-D set, where k varies from 10 to 80 with an interval of 5. Other parameters are fixed as follows: $n = 0.1$M, $\mu = 0.1$, $k_{\max} = 200$, $\min_c = k$, $\max_c = 200$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$.

TABLE III
INFORMATION OF THE FIVE REAL-WORLD NETWORKS USED IN THE EXPERIMENTS

Networks	# Nodes	# Links	AD	D_{max}	# Coms	AO_m
Amazon	0.3M	0.9M	5.53	549	271K	9.566
DBLP	0.3M	1M	6.62	343	13.5K	5.141
YouTube	1.1M	3M	5.27	28.8K	16.4K	4.727
LiveJournal	4M	34.7M	17.35	14.8K	0.66M	8.57
Orkut	3M	117.2M	76.3	33.3K	6.3M	41.073

AD: average degree, Dmax: maximum degree, # Coms: number of ground-truth communities, AO_m: average O_m
M denotes a million, K denotes a thousand

B. Sensitivity Analysis of Parameters p and α in OCLN

As mentioned in Section 3, there are two important parameters (p and α) in the proposed OCLN, where p is used to control the community size and α is a threshold for removing incorrectly identified nodes in a community. In this section, we investigate the influence of p and α on the performance of OCLN in large-scale LFR synthetic networks.

Figure 3 shows the NMI values of the proposed OCLN on the large-scale LFR-N network set by varying the values of p (α is fixed at 0.2) and α (p is fixed at 4). From Fig. 3(a), it can be seen that a small value (e.g., $p = 1$) or a large value (e.g., $p = 7$) is not a suitable setting for OCLN, as this implies that some nodes belonging to the community are not included, or some nodes not belonging to the community are incorrectly added during the expansions. Thus, on the large-scale LFR synthetic networks, the parameter p is set to 4. Regarding the parameter α, it can be observed from Fig. 3(b) that when α is large (e.g., $\alpha = 0.8$), the performance of OCLN deteriorates significantly because a large value of α causes the filtering of a number of nodes in the community, resulting in poor performance. When α is small (e.g., $\alpha = 0.1$), OCLN does not perform well, as a small value of α indicates that only a small number of incorrectly identified nodes are deleted. Therefore, on the large-scale LFR synthetic networks, the parameter α is set to 0.2.

C. Performance on Large-Scale Synthetic Networks

Herein, we evaluate the performance of different algorithms on two groups of large-scale LFR networks.

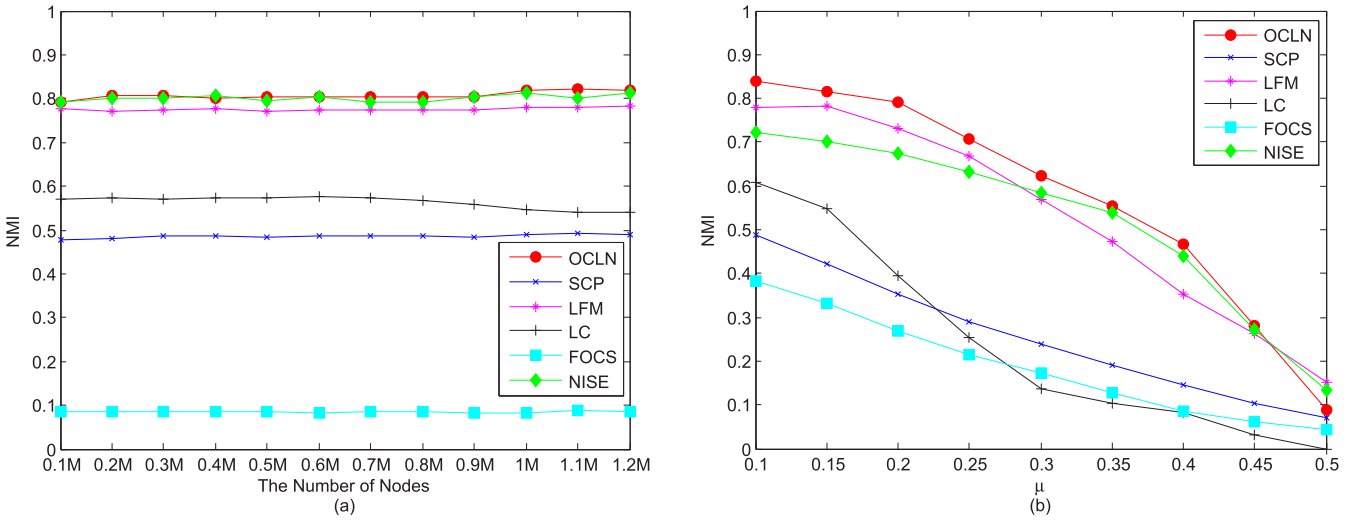

Fig. 5. NMI values of the six algorithms on large-scale LFR-N and LFR-μ sets. (a) NMI on the LFR-N set, where n varies from 0.1M to 1.2M with an interval of 0.1M. Other parameters are fixed as: $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 20$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$. (b) NMI on the LFR-μ set, where μ ranges from 0.1 to 0.5 with an interval of 0.05. Other parameters are fixed as follows: $n = 1.2$M, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$.

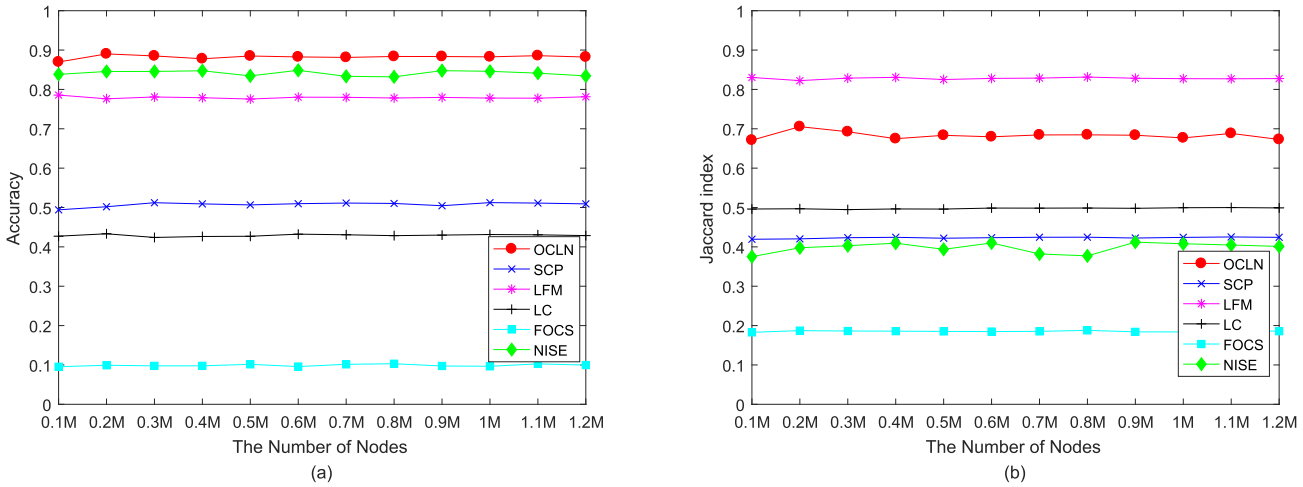

Fig. 6. Accuracy and Jaccard index of the six algorithms on the large-scale LFR-N set. (a) Accuracy on the LFR-N set, where n varies from 0.1M to 1.2M with an interval of 0.1M. Other parameters are fixed as: $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 20$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$. (b) Jaccard index on the LFR-N set, where n varies from 0.1M to 1.2M with an interval of 0.1M. Other parameters are fixed as follows: $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 20$, $o_n = 0.1 \times n$, $o_m = 2$, $\tau_1 = 2$, and $\tau_2 = 1$.

Figure 4 shows the runtime (s) of the proposed OCLN and the other algorithms on the LFR-N and LFR-D network sets. It can be seen that OCLN has smaller runtime than most of the other algorithms. FOCS has the second best performance in terms of runtime on these large-scale LFR sets. The high efficiency of OCLN and FOCS is attributed to the fact that they have linear time complexity with respect to network size. The runtime of LFM, NISE, and LC increases considerably as the network size increases. Therefore, these algorithms are not efficient in handling large-scale networks. The LC algorithm is highly sensitive to the average degree of a network, as shown in Fig. 4(b), and the other algorithms (such as SCP, NISE, FOCS, and OCLN) have almost the same runtime as the average degree increases. Therefore, we can empirically confirm that the proposed OCLN is suitable for large-scale networks in terms of computational efficiency.

Figure 5 shows the NMI values of the six overlapping-community detection algorithms on the LFR-N and LFR-μ sets. From Fig. 5(a), it can be seen that OCLN, NISE, and LFM perform considerably better than LC, SCP, and FOCS on the LFR-N set. The proposed OCLN is slightly better than NISE in terms of NMI, particularly when the network size exceeds 1000000. The LFM algorithm underperforms OCLN and NISE on all considered large-scale LFR networks in the LFR-N set. As shown in Fig. 5 (b), the proposed OCLN algorithm achieves the best NMI values on the LFR-μ set when $\mu < 0.5$; this demonstrates the effectiveness of OCLN on large-scale networks, with different values of the mixing parameter μ, in terms of the quality of the detected communities. From Fig. 5, it can be concluded that the proposed OCLN is superior to the other algorithms in terms of NMI.

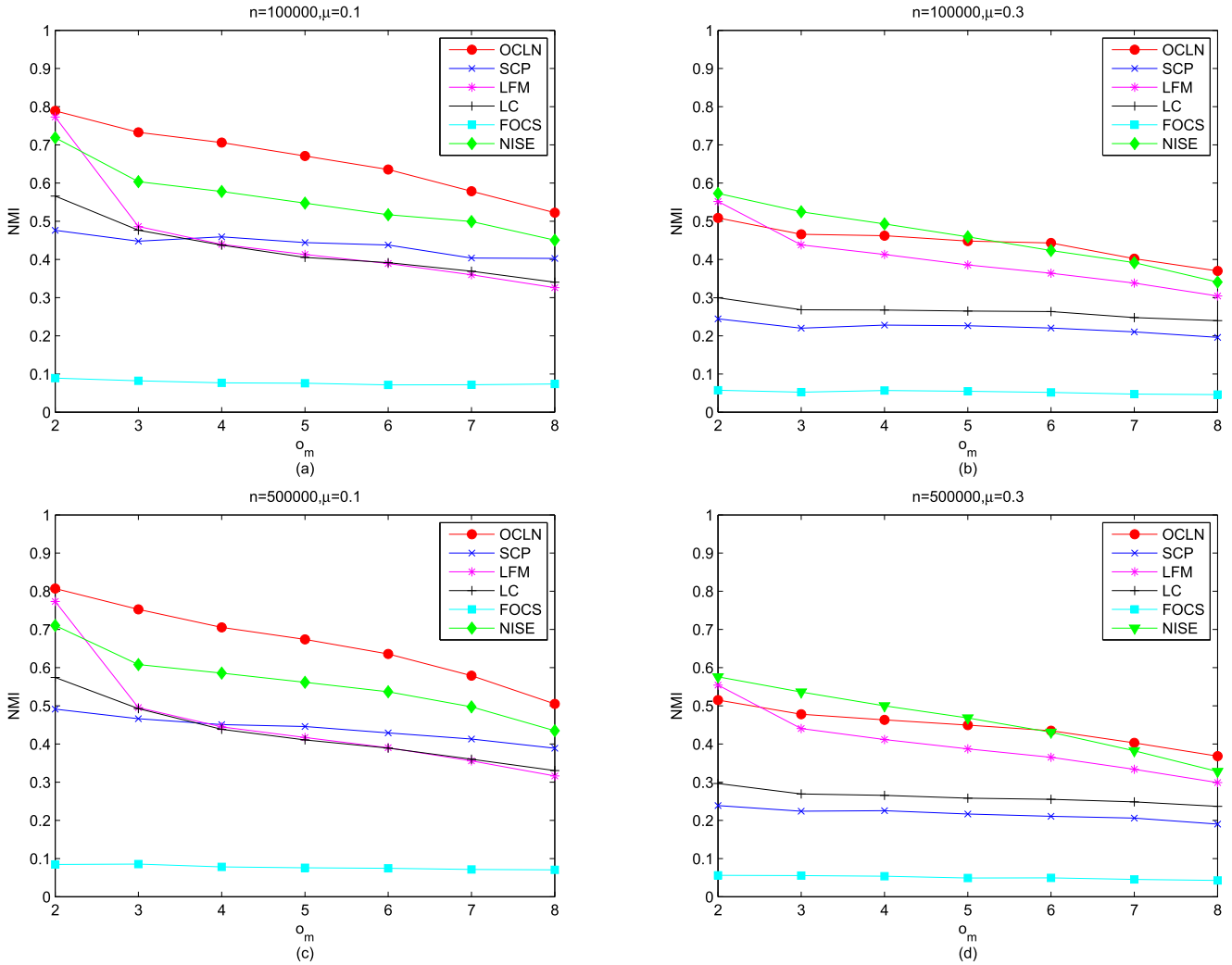

Fig. 7. NMI values of the six algorithms on large-scale LFR synthetic networks with different o_m. (a) NMI on the LFR-Om1 set, where o_m ranges from 2 to 8 with an interval of 1. Other parameters are fixed as follows: $n = 0.1$M, $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $\tau_1 = 2$, and $\tau_2 = 1$, (b) NMI on the LFR-Om2 set, where o_m ranges from 2 to 8 with an interval of 1. Other parameters are fixed as follows: $n = 0.1$M, $\mu = 0.3$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $\tau_1 = 2$, and $\tau_2 = 1$, (c) NMI on the LFR-Om3 set, where o_m ranges from 2 to 8 with an interval of 1. Other parameters are fixed as follows: $n = 0.5$M, $\mu = 0.1$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $\tau_1 = 2$, and $\tau_2 = 1$, and (d) NMI on the LFR-Om4 set, where o_m ranges from 2 to 8 with an interval of 1. Other parameters are fixed as follows: $n = 0.5$M, $\mu = 0.3$, $k = 10$, $k_{\max} = 50$, $\min_c = 20$, $\max_c = 100$, $o_n = 0.1 \times n$, $\tau_1 = 2$, and $\tau_2 = 1$.

Figure 6 shows the accuracy and Jaccard index of different overlapping-community detection algorithms on the LFR-N set. It can be observed that the proposed OCLN always achieves the best or second best performance in terms of both metrics.

Figure 7 shows the NMI values of the six overlapping-community detection algorithms on the LFR-Om1, LFR-Om2, LFR-Om3, and LFR-Om4 sets. It can be seen that the proposed OCLN has competitive performance on all types of LFR networks under different levels of overlapping diversity. For $\mu = 0.1$, the proposed OCLN achieves significantly better NMI values than the other algorithms on LFR networks with overlapping diversity o_m ranging from 2 to 8. The proposed OCLN also has promising performance under different overlapping diversity when $\mu = 0.3$, achieving an NMI value comparable with those obtained by the other algorithms. The proposed OCLN obtains the best NMI value when the overlapping diversity o_m is larger than 5 for $\mu = 0.3$.

In conclusion, the proposed OCLN algorithm is suitable for overlapping-community detection on large-scale synthetic networks, in terms of both computational efficiency and the quality of detected communities.

D. Performance on Large-Scale Real-World Networks

Herein, we evaluate the performance of the six overlapping-community detection algorithms on five large-scale, real-world networks.

Tables IV and V present the runtime and NMI values of the algorithms. The parameter values of all comparison algorithms are set as in the case of the synthetic networks, and the parameters p and α in OCLN are set to $p = 2$ and $\alpha = 0.2$. From the tables, the following two observations can be made.

TABLE IV
RUNTIME(S) OF THE SIX OVERLAPPING COMMUNITY DETECTION ALGORITHMS ON THE FIVE LARGE -SCALE REAL -WORLD NETWORKS

	#Communities/Runtime					
Networks	SCP	LFM	LC	FOCS	NISE	OCLN
Amazon	23.1K/34.513 s	52.5K/23.1 min	34.5K/42.1 s	21.1K/3.76 s	27.8K/1.16 h	19K/**2.6 s**
DBLP	47.3K/32.525 s	64.5K/24.2 min	55.6K/128.7 s	24.2K/**2.71 s**	26.5K/22.4 min	28.6K/3.4 s
YouTube	7.4K/43.5 min	7.4K/10.6 h	0.2K/4.9 h	7.3K/**1.1 min**	30.5K/2.19 h	119.6K/3.2 min
LiveJournal	-	-	-	0.2M/**12.1 min**	-	149.1K/38.2 min
Orkut	-	-	-	0.2M/**1.3 h**	-	14K/3.5 h

h, min, and s denote hour(s), minute(s), and second(s), respectively. M denotes a million, and K denotes a thousand. The blanks in the table denote that the method cannot generate any results within 24 h.

TABLE V
NMI VALUES OF THE SIX OVERLAPPING COMMUNITY DETECTION ALGORITHMS ON THE FIVE LARGE-SCALE REAL-WORLD NETWORKS

Networks	SCP	LFM	LC	FOCS	NISE	OCLN
Amazon	0.2208	0.2074	0.2451	0.2236	0.0969	**0.2525**
DBLP	**0.2196**	0.1289	0.1797	0.2145	0.0704	0.1179
YouTube	**0.0426**	0.0212	0.0052	0.0335	0.0032	0.0390
LiveJournal	-	-	-	0.0307	-	**0.5857**
Orkut	-	-	-	0.0611	-	**0.6102**

The blanks in the table denote that the method cannot generate any results within 24 h.

First, FOCS and OCLN exhibit considerably better computational efficiency than SCP, LFM, LC, and NISE, particularly on the LiveJournal and Orkut datasets, the size of which exceeds 1000000. On these datasets, SCP, LFM, LC, and NISE cannot obtain any result, even after 24 h. The high efficiency of FOCS and OCLN confirms that the local community expansion is a promising strategy for overlapping-community detection in large-scale networks. It should also be noted that although LFM is also based on local community expansion, it is quite time-consuming when the network is large. This is because it needs to repeatedly calculate the local fitness function in each expansion, which takes a significant amount of runtime.

Second, the proposed OCLN achieves a competitive performance on the five large-scale real-world networks in terms of detected-community quality. The proposed OCLN obtains the best NMI value on three networks and the second best on one network. Compared with FOCS, OCLN performs significantly better with respect to the detected-community quality, particularly on the two real-world networks with more than 1000000 nodes, that is, LiveJournal and Orkut. The proposed OCLN achieves an NMI value of 0.5857 and 0.6102 on LiveJournal and Orkut, respectively, whereas the corresponding values for FOCS are only 0.0307 and 0.0611. It appears that OCLN is more suitable for large-scale networks with high overlapping diversity, as Amazon, LiveJournal, and Orkut (on which OCLN achieves the best NMI value) have a relatively high overlapping diversity O_m, as shown in Table III.

From the above observations, we can conclude that compared with the other algorithms, the proposed OCLN can achieve both high community quality and low computational cost; this is highly promising for large-scale, real-world networks, particularly when there are millions of nodes.

V. CONCLUSION

In this paper, we proposed an efficient overlapping-community detection algorithm (OCLN) for large-scale networks. In OCLN, an expansion method based on local-neighborhood information was proposed to achieve fast community expansion. In each expansion, the proposed method only considers the neighbors of nodes added in the last expansion, instead of all neighbors of the current community. Theoretical analysis demonstrated that OCLN has linear time complexity with respect to network size, which is better than most of the existing overlapping community detection algorithms. The performance of OCLN was also verified on large-scale synthetic and real-world networks, demonstrating the superiority of the proposed algorithm over existing state-of-the-arts in terms of both effectiveness and efficiency.

It was demonstrated that using the local-neighborhood information is a promising method for fast overlapping-community detection in large-scale networks. In the future, this technique will be further explored for enhancing the efficiency for large-scale networks. An interesting research direction is to consider the local-neighborhood information in other types of large-scale complex networks, such as signed [62] and dynamic networks [63]. In addition, in this study we only focused on combing the proposed local neighborhood technique with the local-expansion method. However, the combination with other overlapping-community detection methods (such as link community [35] and label propagation [33]) should be further investigated.

REFERENCES

[1] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surveys*, vol. 45, no. 4, pp. 1–35, Aug. 2013.
[2] J. Yuan, Z. Liu, and X. Qiu, "Community detection in complex networks: Algorithms and analysis," in *Proc. Int. Conf. Trustworthy Comput. Services*, 2014, pp. 238–244.
[3] R. Aldecoa and I. Marin, "SurpriseMe: An integrated tool for network community structure characterization using surprise maximization," *Bioinformatics*, vol. 30, no. 7, pp. 1041–1042, Apr. 2014.
[4] H.-J. Li and J. J. Daniels, "Social significance of community structure: Statistical view," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 91, no. 1, Jan. 2015, Art. no. 012801.
[5] A. Sallaberry, F. Zaidi, and G. Melançon, "Model for generating artificial social networks having community structures with small-world and scale-free properties," *Social Netw. Anal. Mining*, vol. 3, no. 3, pp. 597–609, Sep. 2013.
[6] X. Fan Wang and G. Chen, "Complex networks: Small-world, scale-free and beyond," *IEEE Circuits Syst. Mag.*, vol. 3, no. 1, pp. 6–20, Sep. 2003.
[7] H.-J. Li, H. Wang, and L. Chen, "Measuring robustness of community structure in complex networks," *EPL (Europhysics Letters)*, vol. 108, no. 6, p. 68009, Dec. 2014.

[8] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2282–2290.

[9] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 261–277, Feb. 2014.

[10] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.

[11] Y. W. Jiang, C. Y. Jia, and Y. U. Jian, "Overlapping community detection in complex networks based on cluster prototypes," *Pattern Recognit. Artif. Intell.*, vol. 26, no. 7, pp. 648–659, 2013.

[12] 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, pp. 2142–2152, Nov. 2009.

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

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

[15] 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.

[16] X. Yu, J. Yang, and Z.-Q. Xie, "A semantic overlapping community detection algorithm based on field sampling," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 366–375, Jan. 2015.

[17] X. Wang, L. Jiao, and J. Wu, "Adjusting from disjoint to overlapping community detection of complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 24, pp. 5045–5056, Dec. 2009.

[18] 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.

[19] 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.

[20] Z. Kuncheva and G. Montana, "Community detection in multiplex networks using locally adaptive random walks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2015, pp. 1308–1315.

[21] W. Wang, D. Liu, X. Liu, and L. Pan, "Fuzzy overlapping community detection based on local random walk and multidimensional scaling," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 24, pp. 6578–6586, Dec. 2013.

[22] B. Cai, H. Wang, H. Zheng, and H. Wang, "An improved random walk based clustering algorithm for community detection in complex networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2011, pp. 2162–2167.

[23] J. Q. Jiang, A. W. M. Dress, and G. Yang, "A spectral clustering-based framework for detecting community structures in complex networks," *Appl. Math. Lett.*, vol. 22, no. 9, pp. 1479–1482, Sep. 2009.

[24] R. Langone, C. Alzate, and J. A. K. Suykens, "Kernel spectral clustering for community detection in complex networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–8.

[25] H.-J. Li, Y. Wang, L.-Y. Wu, Z.-P. Liu, L. Chen, and X.-S. Zhang, "Community structure detection based on potts model and network's spectral characterization," *EPL (Europhysics Letters)*, vol. 97, no. 4, p. 48005, Feb. 2012.

[26] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, p. 814, Jun. 2005.

[27] 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, Aug. 2008, Art. no. 026109.

[28] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

[29] 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.

[30] E. Gregori, L. Lenzini, and S. Mainardi, "Parallel (k)-clique community detection on large-scale networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1651–1660, Aug. 2013.

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

[32] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, pp. 2011–2024, Oct. 2010.

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

[34] L. Huang, G. Wang, Y. Wang, E. Blanzieri, and C. Su, "Link clustering with extended link similarity and EQ evaluation division," *PLoS ONE*, vol. 8, no. 6, Jun. 2013, Art. no. e66005.

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

[36] 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, Jul. 2009, Art. no. 016105.

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

[38] J. Leskovec and A. Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. Accessed: Jun. 2014. [Online]. Available: http://snap.stanford.edu/data

[39] 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.

[40] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 2099–2108.

[41] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, pp. 155–168, 2008.

[42] T. Zhang and B. Wu, "A method for local community detection by finding core nodes," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 1171–1176.

[43] Q. Chen, T.-T. Wu, and M. Fang, "Detecting local community structures in complex networks based on local degree central nodes," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 3, pp. 529–537, Feb. 2013.

[44] H.-J. Li, Z. Bu, A. Li, Z. Liu, and Y. Shi, "Fast and accurate mining the community structure: Integrating center locating and membership optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2349–2362, Sep. 2016.

[45] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, pp. 19–44, 2009.

[46] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1272–1284, May 2016.

[47] J. Huang, H. Sun, Y. Liu, Q. Song, and T. Weninger, "Towards online multiresolution community detection in large-scale networks," *PLoS ONE*, vol. 6, no. 8, Aug. 2011, Art. no. e23829.

[48] F. Moradi, T. Olovsson, and P. Tsigas, "A local seed selection algorithm for overlapping community detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 1–8.

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

[50] Y. K. Shih and S. Parthasarathy, "Identifying functional modules in interaction networks through overlapping Markov clustering," *Bioinformatics*, vol. 28, no. 18, pp. 473–479, 2012.

[51] C. Pizzuti and S. E. Rombo, "Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods," *Bioinformatics*, vol. 30, no. 10, pp. 1343–1352, May 2014.

[52] X. Wen, W. Chen, Y. Lin, and T. Gu, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, pp. 1–14, 2016.

[53] T. Chakraborty, S. Kumar, N. Ganguly, A. Mukherjee, and S. Bhowmick, "GenPerm: A unified method for detecting non-overlapping and overlapping communities," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2101–2114, Aug. 2016.

[54] 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. (IADIS)*, 2005, pp. 97–104.

[55] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Seeding for pervasively overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 6, Jun. 2011, Art. no. 066107.

[56] U. Brandes, "A faster algorithm for betweenness centrality," *J. Math. Sociol.*, vol. 25, no. 2, pp. 163–177, 2001.
[57] M. Kitsak *et al.*, "Identification of influential spreaders in complex networks," *Nature Phys.*, vol. 6, no. 11, p. 888, 2010.
[58] C. Castellano, F. Cecconi, V. Loreto, D. Parisi, and F. Radicchi, "Self-contained algorithms to detect communities in networks," *Eur. Phys. J. B, Condens. Matter*, vol. 38, no. 2, pp. 311–319, Mar. 2004.
[59] H. Liu, Z. Wu, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.
[60] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 1151–1156.
[61] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, Oct. 2008, Art. no. 046110.
[62] 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, p. 2274, Dec. 2014.
[63] K. Kim, R. McKay, and B.-R. Moon, "Multiobjective evolutionary algorithms for dynamic social network clustering," in *Proc. 12th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Portland, OR, USA, Jul. 2010, pp. 1179–1186.

Fan Cheng received the B.Sc. and M.Sc. degrees from the Hefei University of Technology, China, in 2000 and 2003, respectively, and the Ph.D. degree from the University of Science and Technology of China, China, in 2012.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, China. He has published more than 40 papers in refereed conferences and journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION (TEVC), IEEE TRANSACTIONS ON BIG DATA (TBD), IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING (TNSE), and *IEEE Computational Intelligence Magazine* (CIM). His main research interests include machine learning, imbalanced classification, multi-objective optimization, and complex networks.

Congtao Wang received the B.Sc. and M.S. degrees from the School of Computer Science and Technology, Anhui University, Hefei, China, in 2014 and 2017, respectively. His current research interests include complex networks and data mining.

Xingyi Zhang (Senior Member, IEEE) received the B.Sc. degree from the 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.

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, evolutionary multi-objective optimization, and complex network analysis. He has published more than 90 papers in refereed conferences and journals, such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION (TEVC), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), *IEEE Computational Intelligence Magazine* (CIM), and *Information Sciences*. He was a recipient of the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award.

Yun Yang received the B.Sc. degree (Hons.) in information technology and telecommunication from Lancaster University, Lancaster, U.K., in 2004, the M.Sc. degree in advanced computing from Bristol University, Bristol, U.K., in 2005, and the M.Phil. degree in informatics and the Ph.D. degree in computer science from the University of Manchester, Manchester, U.K., in 2006 and 2011, respectively.

He was a Research Fellow with the University of Surrey, Surrey, U.K., from 2012 to 2013. He is currently with the National Pilot School of Software, Yunnan University, Kunming, China, as a Full Professor of machine learning. His current research interests include machine learning, data mining, pattern recognition, and temporal data process and analysis.