A Novel Method for Detecting New Overlapping Community in Complex Evolving Networks

Jiujun Cheng, Xiao Wu, Mengchu Zhou, *Fellow, IEEE*, Shangce Gao, *Senior Member, IEEE*, Zhenhua Huang, and Cong Liu

***Abstract*—It is an important challenge to detect an overlapping community and its evolving tendency in a complex network. To our best knowledge, there is no such an overlapping community detection method that exhibits high normalized mutual information (NMI) and *F*-score, and can also predict an overlapping community's future considering node evolution, activeness, and multiscaling. This paper presents a novel method based on node vitality, an extension of node fitness for modeling network evolution constrained by multiscaling and preferential attachment. First, according to a node's dynamics such as link creation and destruction, we find node vitality by comparing consecutive network snapshots. Then, we combine it with the fitness function to obtain a new objective function. Next, by optimizing the objective function, we expand maximal cliques, reassign overlapping nodes, and find the overlapping community that matches not only the current network but also the future version of the network. Through experiments, we show that its NMI and *F*-score exceed those of the state-of-the-art methods under diverse conditions of overlaps and connection densities. We also validate the effectiveness of node vitality for modeling a node's evolution. Finally, we show how to detect an overlapping community in a real-world evolving network.**

***Index Terms*—Evolving network, fitness function, maximal clique, multiscaling, node fitness, node vitality, overlapping community, shared community degree.**

Manuscript received August 7, 2017; accepted November 17, 2017. This work was supported in part by the National Natural Science Foundation of China (Key Program) under Grant 61331009, in part by the National Natural Science Foundation of China under Grant 61472284 and Grant 61772366, in part by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant G-415-135-38, and in part by JSPS KAKENHI under Grant JP17K12751. This paper was recommended by Associate Editor E. Herrera-Viedma. *(Corresponding authors: Mengchu Zhou; Shangce Gao; Zhenhua Huang.)*

J. Cheng, X. Wu, and Z. Huang are with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China (e-mail: chengjj@tongji.edu.cn; smfwuxiao@163.com; huangzhenhua@tongji.edu.cn).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Renewable Energy Research Group, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: zhou@njit.edu).

S. Gao is with the Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan (e-mail: gaosc@eng.u-toyama.ac.jp).

C. Liu is with the Shandong University of Science and Technology, Qingdao 266590, China (e-mail: liucongchina@sdust.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org provided by the authors. This includes a PDF containing additional figures, tables, and algorithms relevant to the paper. This material is 0.128 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2779138

I. Introduction

MANY systems can be represented as networks, in which nodes stand for individuals and links for their interactions. Such networks are useful. For example, Liu and Jia [1] proposed a new algorithm for evaluating trust level in service-oriented online social networks. Yang *et al.* [2] utilized location-based social networks to model and infer user activity preferences. Eustace *et al.* [3] performed community detection to approximate Web communities from the topic related Web pages based on subspace factorization. Until now, researchers have investigated these networks and found important attributes like the power-law degree distribution, small-world phenomenon [4], and preferential attachment [5]. Among them, the *community structure* [6] provides a profound insight into these systems and is useful for understanding the organizations and revealing hidden relationships among nodes. Many community detection methods were proposed, which fall into partitioning approaches [6], density-based methods [7], clique-based methods [8], [9], hierarchical clustering methods [10], [11], and statistical-based methods [12].

However, the community structures detected by these methods are isolated from each other and cannot share common nodes. In a real-world network, different communities often overlap. To detect an overlapping community, Palla *et al.* [8] proposed the clique percolation method (CPM). However, recent studies demonstrated its low accuracy reflected by normalized mutual information (NMI). CPM just considers a local area of the group when it is conducting k-clique template rolling, which can loosen the overall connectivity among all nodes in the group. The Lancichinetti–Fortunato method (LFM) [13] generalizes the original k-clique template rolling to the greedy optimization of an objective function, which considers not only local area but also the overall one. But it is susceptible to the quality of a seed, namely the starting point for expansion. By using maximal cliques, greedy clique expansion (GCE) [14] obtains a substantial improvement over LFM. Recently, Moradi *et al.* [15] proposed a seed selection algorithm to improve overlapping community detection.

In static networks, the overlapping community detection has been extensively explored since the work [8]. However, most of the real-world networks evolve. In routing analysis about Internet of Vehicles [16], for example, vehicle nodes join or leave groups and move from one group to another. Their activeness changes randomly. The dynamics

of evolving networks are increasingly attracting researchers' attentions. For example, the work in [17] explores mesoscopic patterns of human interaction networks using digital traces of today's human interactive activities. Group evolution is investigated in [18]. Chakrabarti *et al.* [19] introduce an evolutionary clustering framework, which makes clusterings between consecutive networks smooth while at the same time preserving high snapshot quality. Their framework is later extended in [20]–[23]. However, what they have accomplished is answering how to make clustering results between consecutive networks smooth rather than detecting predictable communities. More recently, community detection in dynamic networks is improved in [24]. A new measure for an overlapping community called *compactness* is proposed in [25]. Currently, there is no ideal method for predicting the future status of an overlapping community despite its usefulness. For example, the epidemic spreading research can benefit from overlapping community prediction in time-varying networks [26], as a community structure has a significant impact on the susceptible–infected–susceptible process. Knowing the community in advance helps one design a desired policy to suppress the propagation of an infectious disease. We suggest expanding the clique-expansion-based method for this task because it can detect overlapping communities well, and is easy to incorporate nontopological factors. Other methods cannot detect overlapping communities, or have too complex models to be extended, therefore, they are unsuitable for community prediction. Several new and important problems arise when the existing clique-expansion-based methods are adopted for overlapping community prediction.

1) In evolving networks, fewer nodes are active while more are inert, and node's activeness changes with the network evolution. Existing methods treat nodes equally and cannot be used to predict future communities [27]. Fig. S1 in the supplementary material illustrates heterogeneous activeness distribution of nodes.
2) When connection densities in communities are weak (but stronger than average), the expansion-based methods fail because the seeds tend to expand excessively. That is because the probability of each node to be chosen among the neighbors of the community are close, and therefore may lead to improper choices.
3) Existing methods detect communities without consideration of assigning overlapping nodes among communities properly, thereby it causes the overlapping nodes' misassignment problem [27].

To solve these problems, we first propose the concept of node vitality for each node's activeness by taking heterogeneous activeness distribution into account. Then, by studying the traditional fitness function under the stochastic blockmodel and combining it with node vitality, we formulate a new function, and use it as an optimization objective. Based on these, we present a new overlapping community detection method. It not only reveals relationships among nodes in the current snapshot but also has a high probability to match the corresponding node groups in the future. Finally, with regard to the misclassification of the overlapping nodes, we propose a reassigning algorithm to further enhance our proposed method.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III introduces the concept of node vitality, which represents intrinsic activeness of a node. Then, the fitness function that combines node vitality is presented. Finally, a clique expansion-based algorithm is given. Section IV discusses how to solve the misclassification problem for overlapping nodes. In Section V, the experimental results on both synthetic and empirical data sets are given to verify the proposed method. Section VI draws the conclusion.

II. Related Work

We summarize the existing work related to node fitness for multiscaling in an evolving network and expansion objective for overlapping community detection in a static network. All of the symbols used in this paper are summarized in Table S1 in the supplementary material.

A. Node Fitness

To model the fact that new nodes have different evolution speeds, Bianconi and Barabási [29] propose the concept of *node fitness*, which is a parameter assigned to each node to measure its intrinsic ability to compete for links in evolving networks. As a node's speed in creating links depends on its degree, node fitness influences its speed as well. The phenomenon that nodes have different capabilities to attract other nodes is called *multiscaling*, which is witnessed in many real networks. In social networks, for example, some people having only a few friends at first, may make friends with others in a short time due to their good social skills. Besides multiscaling, *preferential attachment* [5] is widely accepted to model a node's evolution, which says that a high-degree node is more likely to create new connections with others. It is also known as the "richer get richer" phenomenon. Note that the *node fitness* here is the attribute of a node. It has nothing to do with a fitness function to be described later.

According to [29], *node fitness* $\eta \in R$ is usually chosen by following some distribution, where R denotes the set of real numbers. Currently, the exact value range for η is unclear yet. In an evolving network G, the edge probability between node i and other nodes is proportional to the product of degree k_i and node fitness η_i, which can be measured as

$$p(i) = \frac{\eta_i k_i}{\sum_{j \in G} \eta_j k_j}. \tag{1}$$

The denominator in (1) is for the entire network. Then, the speed at which node i changes its connectivity can be derived as follows:

$$\frac{\partial k_i}{\partial t} = w \cdot p(i) = \frac{w \eta_i k_i}{\sum_{j \in G} \eta_j k_j} \tag{2}$$

where w is the total number of new links in the network divided by the number of new nodes, which accounts for the

fact that each new node adds w links on average [29]. The product of factor w, current degree k_i and its fitness η_i can effectively express each individual's ability to compete for new links. However, the drawbacks of node fitness η_i are obvious. First, it is defined to be a fixed attribute of a node, inconsistent with real-world observations, so that the consideration of evolution by $p(i)$ is limited. Second, it characterizes nodes just in growing networks and neglects contracting ones. Finally, a person cannot obtain the exact node fitness value η_i.

B. Expansion Objective

For an expansion-based overlapping community detection method, a fitness function f takes a set of nodes from graph G and returns a value which indicates how densely connected these nodes are. The values returned by f usually correspond to some definition of communities. The idea of detecting local communities by optimizing a fitness function as an optimization objective was applied in the studies [30]–[32], in which the concept of an overlapping community is formalized in different ways.

Although multiple fitness functions have been proposed, the following function [13] is widely accepted to give good results in various types of networks as

$$f = \frac{W_{\text{in}}}{(W_{\text{in}} + W_{\text{out}})^{\alpha}} \tag{3}$$

where W_{in} and W_{out} are the sums of internal degrees and external degrees respectively, and α governs the resolution of a detected overlapping community. A higher α yields smaller communities, while a lower α yields larger communities. The work in [14] has demonstrated that α in range [0.9, 1.5] gives the best performance. W_{in} can be interpreted as the link connectivity density of the group, and W_{out} is the number of links connecting between the current group and remaining part of a network. When α is given, a good community should have a large W_{in} value and a low W_{out} value.

The described fitness function is the most important concept of many expansion-based algorithms, such as the prior mentioned LFM [13] and GCE [14]. However, the intrinsic equality in treating each node has resulted in the symmetrical expansion, which is to some extent deviated from real evolving networks. This drawback can also be explained in another aspect. It does not take heterogeneous degree distribution or multiscaling into account because (3) just aggregates over all nodes in the group, thus having simple aggregate sums W_{in} and W_{out} used to evaluate an overlapping community. One might claim that this shortage has limited influences. But when a community detection algorithm is run on graphs with weak communities, e.g., on the Lancichinetti–Fortunato–Radicchi (LFR) graph [33] with its parameter $\mu > 0.6$, optimizing (3) can never choose proper nodes, thus leading to an excessive expansion problem.

C. Summary

Recently, research into evolving networks has drawn much public attention and enjoyed an accelerated flush, especially in the area of dynamic overlapping community detection. However, more efforts are required to detect overlapping communities and improve NMI and F-score when facing a dynamic topology and excessive expansion problem.

III. Asymmetric Clique Expansion With Node Vitality

In this section, we describe the most essential components of our proposed asymmetric clique expansion with node vitality (ACENV). First, we propose the concept of node vitality to quantify a node's activeness. Second, we discuss how to apply it to the expansion optimization objective. Next, we discuss the details of using cliques as seeds for expansion. Finally, we present a complete procedure of ACENV and point out the differences when compared with other existing expansion-based methods.

A. Node Vitality

In real-world networks, each node has different activeness and the number of active nodes is usually much less than that of the inactive ones. A node is said to be *active* if its activeness is high. For example, in a social network like Facebook or Twitter, only a small percent of users are active while others are relatively quiet. We call this phenomenon a heterogeneous activeness distribution. Evolving networks are frequently characterized by the co-existence of active and inactive nodes. Fig. S1 in the supplementary material illustrates the heterogeneous activeness distribution phenomenon in a real-world network.

To measure the activeness of each node, we introduce *node vitality* by extending *node fitness*, which quantifies the ability to compete for new links [29], but is a fixed value for each node. We define the vitality to be a real number $v_i \in [-1, 1]$ associated with node i, and it indicates the intrinsic ability of node i to create or remove links. For example, in social networks, it may reflect an individual's skill of building connections with others in a short time. When $v_i > 0$, node i has a strong desire to create more links, while $v_i < 0$ indicates that it is going to remove connections. A large $|v_i|$ suggests that the corresponding node is behaving actively and has a high probability of changing its community memberships in the future, while a low $|v_i|$ suggests that the corresponding node tends to be unchanged.

In the process of network evolution, the vitality of a node influences its link changes, thus causing its degree k_i to change. We can obtain a node's vitality v_i by comparing g_{t-1} and g_t and leveraging degree k_i. We assume that an evolving network is represented as a sequence of network snapshots $\mathcal{G} = \{g_1, \ldots, g_n\}$ at some reasonable time intervals. Each network snapshot g_t is referred to as $g_t(V_t, E_t)$ $(1 \le t \le n)$, where V_t and E_t are the sets of nodes and edges in g_t. Similar to η_i, node vitality v_i is also related to node degree's change speed. We derive v_i from k_i and η_i. Node fitness η_i for node i usually conforms to some distribution $\rho(\eta)$, but each node's fitness value and the distribution are unknown. Suppose that when the Bernoulli distribution is conformed, namely all node fitness values are equal to a value, we can obtain a node degree's evolution as a function of time $k_i(t) \sim t^{1/2}$ from (2).

This is equivalent to the scale-free model with the preferential attachment [5] but without the aforementioned multiscaling. When another distribution is assumed, the degree evolution of k_i is governed by a dynamic exponent $\beta(\eta_i)$ as follows if k_i is assumed to follow a power law

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\beta(\eta_i)} \tag{4}$$

where $k_i(t)$ is a time-dependent function of degree of node i, t is the current age of a network, t_i is the age of node i, m is the total number of created or removed links divided by the number of added or removed nodes, and $\beta(\eta_i) \in (0, 1)$ is a bounded exponent, which is influenced by the distribution $\rho(\eta)$ and node fitness η_i. As mentioned above, it is insufficient for node fitness η_i to capture a node's dynamics. We reformulate $\beta(\eta_i)$ to $\alpha(v_i(t))$ in (4), in which α is a node vitality-dependent function for the dynamic exponent. v_i is usually in the positive relationship with η_i in a single snapshot. For simplicity, we assume α to be a constant function because it provides the positive relationship, namely, $v_i(t) = \alpha^{-1}(\beta(\eta_i)) = u{\cdot}\beta(\eta_i)$. Here, u is neither relevant to node fitness η_i nor β, so it can be safely ignored. Note that (4) fails to consider that some nodes may remove links ($k_i < 0$) while the network is growing namely $m > 0$. Based on these observations, we assume $v_i(t) = \beta(\eta_i)$ in one specific snapshot. Based on this assumption and (4), node vitality $v_i(t)$ for node i in graph g_t can be computed as

$$v_i(t) = \operatorname{sgn}(k_i(t) - m) \cdot \frac{\log|k_i(t)/m|}{\log(t/t_i)} \tag{5}$$

where $m \neq 0$ must be assured and age of the network t is larger than that of any node t_i. The sign function $\operatorname{sgn}(\cdot)$ is introduced to preserve the evolution direction. In (5), the node degree's changing rate $k_i(t)$ can be approximated by $k_i(t) \approx k_i^t - k_i^{t-1}$, i.e., the changed node degree compared with the previous network snapshot. Then $v_i(t)$ in snapshot g_t by comparing with g_{t-1} can be formulated as

$$v_i(t) = \operatorname{sgn}\left(k_i^t - k_i^{t-1} - m\right) \cdot \frac{\left|\log\left|\left(k_i^t - k_i^{t-1}\right)/m\right|\right|}{\log(t/t_i)}. \tag{6}$$

When $k_i^t = k_i^{t-1}$ happens for node i, we define $v_i(t) = -\infty$ according to (5). Note that the age of the network is always larger than a node's age, namely, $t > t_i$.

Equation (6) is not applicable to a static network, but it is useful for changing networks. The nodes with high $|v_i|$ are active because large nonzero $|v_i|$ values indicate that they are quickly changing their connections, and the nodes with zero or small $|v_i|$ are inactive. Nodes with $v_i = 0$ are completely inactive since their structures are unchanged. When ACENV is applied to network $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$, each snapshot is iterated, and the current snapshot's index $\tau \in \{1, 2, \ldots, n\}$ is adopted as network age. We also assume that each node appears in the network at different time. By comparing between network snapshots g_t and g_{t-1} in (6), larger vitality values to nodes which are building or losing more links is assigned. However, (6) may get a value out of the range $(-1, 1)$, and because we define v_i to be in the range $(-1, 1)$, the final node vitality should also be normalized via the following sigmoid function:

$$v_i(t) = \frac{2}{1 + e^{-v_i(t)}} - 1. \tag{7}$$

To find out the active nodes and analyze them, we introduce a specific node vitality threshold $v^* \in [0, 1)$. Thus, the nodes whose vitality values $|v_i| > v^*$ are viewed as active and others are inactive. To find a proper threshold, one of the straightforward ways is to use the average node vitality over all nodes. It is formulated as

$$v^* = \frac{1}{n} \cdot \sum_{j \in G} |v_j|. \tag{8}$$

However, this approach may not be appropriate for some special situations. The network analysts need to adjust the approach according to the their specific situations.

B. Fitness Function With Evolution Trends

The fitness function is used to measure the "seniority" of a set of nodes, so that a high fitness means a good community. It is the most important component of an expansion-based method. Traditional fitness functions assume that the activeness of all nodes are equal by ignoring their different evolution trends, and therefore making the results deviate from the clustering in the following snapshot of the sequence. To account for a node's evolution, we introduce a new parameter ρ for the evolution tendency, which is the similarity between a community in g_t and its evolved incarnation in g_{t+1}. An overlapping community should be characterized by dense internal connections, sparse external connections, and high evolution similarity. To detect such overlapping community, we apply node vitality as a fitness function by combining (3) and evolution similarity, and then use the derived function as a new objective function.

With the help of stochastic block-model [34], we first explain how the traditional fitness function in (3) is derived. The stochastic block-model is a popular tool for detecting communities and synthesizing networks. In its simplest form, there are K blocks, groups, or communities and each node belongs to one of the blocks. Then, a $K \times K$ probability matrix ψ can be defined such that each matrix element $\psi_{i,j}$ is the link probability between a node in block i and another one in block j. The model assumes that a network is generated by matrix ψ. Therefore, the expansion-based community detection method is equivalent to the process of fitting a stochastic block-model with a network snapshot, namely inferring model parameters K and ψ by building the blocks. But different from other methods, the expansion methods usually construct each block by processing seeds individually, therefore we need to inspect just one block before proceeding to the next one. For convenience, we develop a simplified variant of the stochastic block-model, which is focused on one block. In our simplified block-model, there are only two blocks: 1) the community block $\mathcal{B}_1$ that corresponds to the partial community currently being expanded and 2) the periphery block $\mathcal{B}_2$ that corresponds to the remaining nodes. Each node can be assigned to block $\mathcal{B}_1$ or $\mathcal{B}_2$, and

is also associated with a node vitality v_i. Block $\mathcal{B}_1$ is initialized with a seed, and will be the overlapping community to be detected after the expansion completes. Then, the matrix ψ can be reduced to a 2×2 link probability matrix:

$$\psi = \begin{pmatrix} \xi_{11} & \xi_{12} \\ \xi_{12} & \xi_{22} \end{pmatrix} \tag{9}$$

in which ξ_{11} is the link probability between any pair of nodes both in block $\mathcal{B}_1$, and ξ_{12} is the link probability between a node in block $\mathcal{B}_1$ and another node in $\mathcal{B}_2$. The matrix element ξ_{22} is the link probability between two nodes in $\mathcal{B}_2$. ξ_{11} describes the connection density in block $\mathcal{B}_1$, while ξ_{12} describes the density across $\mathcal{B}_1$ and $\mathcal{B}_2$. The definition of a community requires that the internal density ξ_{11} should be high while the external density ξ_{12} should be low. The ratio ξ_{11}/ξ_{12} can naturally be used to measure a community's validity. In this case, a higher ξ_{11}/ξ_{12} indicates a closer community.

The model parameters K, ξ_{11} and ξ_{12} and nodes' assignment layout are unknown, and they must be estimated. When a specific network snapshot is given and each pair of nodes are assumed to be connected with identical probability, the model parameter ξ_{11} can be estimated as

$$\xi_{11} = E(p_{ij}) \approx \frac{W_{\text{in}}}{n^2} \tag{10}$$

in which i and j stand for nodes in $\mathcal{B}_1$, $E(p_{ij})$ stands for the expectation of link probability p_{ij} between i and j, and n is the number of nodes in $\mathcal{B}_1$. Equation (10) is the observed links divided by the number of all possible links. Similarly, ξ_{12} can be estimated as:

$$\xi_{12} = E(p_{ij}) \approx \frac{W_{\text{out}}}{nm} \tag{11}$$

in which m is the number of nodes in $\mathcal{B}_2$, i and j are a node in $\mathcal{B}_1$ and $\mathcal{B}_2$ respectively. Since m is usually much larger than n, which leads to a very small nm, we should use n^2 instead of nm in (11), because the probabilities of nodes j in block $\mathcal{B}_2$ linking to a node i in $\mathcal{B}_1$ are not uniformly distributed. In other words, not all nodes in block $\mathcal{B}_2$ are linked to nodes in $\mathcal{B}_1$ with equal probability. Afterwards, the ratio ξ_{11}/ξ_{12} with these estimated parameters can measure an overlapping community as well as being used as the objective function. Using ξ_{11}/ξ_{12} as the expansion objective is a natural solution. A better way is to put the ratio in an increasing function:

$$f(x) = \frac{x}{(1+x)^{\alpha}} = \frac{\xi_{11}/\xi_{12}}{(1+\xi_{11}/\xi_{12})^{\alpha}}. \tag{12}$$

This makes the fitness function more controllable, and when $\alpha = 1$ it is equivalent to (3).

To apply node vitality into the fitness function, we define a new parameter ρ_i in our model. ρ_i describes the probability of a node currently in $\mathcal{B}_1^t$ or $\mathcal{B}_2^t$ and will appear in block $\mathcal{B}_1^{t+1}$. According to this concept, each node's ρ_i can be formally defined as

$$\rho_i = p\left(i \in \mathcal{B}_1^{t+1} \cap \mathcal{B}_1^t\right) + p\left(i \in \mathcal{B}_1^{t+1} \cap \mathcal{B}_2^t\right) \tag{13}$$

in which $p(i \in \mathcal{B}_1^{t+1} \cap \mathcal{B}_1^t) = p(i \in \mathcal{B}_1^t) \cdot p(i \in \mathcal{B}_1^{t+1} | i \in \mathcal{B}_1^t)$ is the probability that a node i will appear in $\mathcal{B}_1^{t+1}$ under the condition of $i \in \mathcal{B}_1^t$, and $p(i \in \mathcal{B}_1^{t+1} \cap \mathcal{B}_2^t) = p(i \in \mathcal{B}_2^t) \cdot p(i \in \mathcal{B}_1^{t+1} | i \in \mathcal{B}_2^t)$ is the probability of $i \in \mathcal{B}_1^{t+1}$ under the condition $i \notin \mathcal{B}_1^t$. Similar to ξ_{11} and ξ_{12}, ρ_i is also unknown and must be approximated from the network snapshot. We assume that this "future membership" probability of a node is proportional to the node vitality v_i and its current connection with $\mathcal{B}_1$ versus the connection with $\mathcal{B}_2$, i.e., $k_{\text{in}}/(k_{\text{in}} + k_{\text{out}})$. As the growth of node degree conforms to a power law against network age t [11] and ρ_i is also relevant to the growth, ρ_i should be in the form of power law, i.e., we have

$$\rho_i \approx k_{\text{in}} \cdot \left(\frac{t}{t_i}\right)^{v_i} \tag{14}$$

where k_{in} is the number of node i's connections with nodes in $\mathcal{B}_1$, t_i is the age of node i, and t stands for the age of next network snapshot. In fact, (14) is derived from (4) by changing the bounded exponent $\beta(\eta_i)$ to node vitality v_i, $k_i(t)$ to our proposed parameter ρ_i, and m to k_{in}.

When ρ_i for each node is defined, the similarity between $\mathcal{B}_1^t$ and $\mathcal{B}_1^{t+1}$ in g_{t+1} can be defined as

$$\rho = \frac{1}{n} \sum_{i \in \mathcal{B}_1^t} \rho_i \tag{15}$$

where n is the number of nodes in $\mathcal{B}_1^t$.

In (3), W_{in} is equal to twice the number of links whose source and target nodes are in $\mathcal{B}_1$ and W_{out} is equal to the number of outgoing links of $\mathcal{B}_1$. But the observed values W_{in} and W_{out} fail to consider the heterogeneous activeness distribution in an evolving network. To take the heterogeneous activeness of nodes into account, we combine the fitness function in (3) with estimated evolutionary similarity ρ for block $\mathcal{B}_1$ to define a new fitness function:

$$f = (1-\beta) \cdot \frac{W_{\text{in}}}{(W_{\text{in}} + W_{\text{out}})^{\alpha}} + \beta \cdot \rho \tag{16}$$

where the control parameter $\beta \in [0, 1]$ is introduced to give a tradeoff between connection density and nodes' evolutionary trends. Now (16) uses not only connection patterns in current network snapshot g_t but also evolution information, and the excessive expansion is depressed by the evolutionary similarity. In the previous expansion-based methods, only the optimization of f in (12) is considered.

The new fitness function in (16) is adopted as our objective function instead of (3) and (12). It treats group members differently, and assigns higher probabilities for active nodes to be assigned to the currently being expanded community, and also promotes the probability of closest nodes in distance measured by k_{in}. By using this new objective function as the greedy optimization objective, the dynamic information are fully considered. In addition, the new fitness function also eliminates the excessive expansion problem. When the number of current members of a group comes close to that of a real community, the number of its neighbor nodes is large and they are not quite distinguishable via (3). At this point, the expansion algorithm using (3) chooses random nodes, and finds more and more neighbors, if the convergence condition is unsatisfied. From (16), the neighbors are distinguishable and

the convergence satisfies the above evolution similarity:

$$\rho = \frac{1}{n} \sum_{i \in \mathcal{B}_1^t} k_{\text{in}} \left(\frac{t}{t_i} \right)^{v_i} \tag{17}$$

while (3) cannot. In this way, the excessive expansion problem is solved.

C. Seeds and Asymmetric Expansion

The choice of seeds for expansion is extremely important as the expansion-based community detection methods heavily depend on the quality of seeds. Some methods use random edges [31] or random nodes [13] as seeds, resulting in relatively poor performances on the LFR benchmarks. Recently, the use of maximal cliques as starting points has shown great performance [14], [35], [36]. In [35] and [36], for example, the maximal cliques, i.e., the maximal complete subgraphs, are detected and merged according to some belonging degree as defined in [35]. We also use maximal cliques as seeds because they are the tightest structures, and their inner connections are the densest in the network.

Before expansion, we need to detect all the maximal cliques in the current snapshot g_t. The problem of enumerating all the cliques has been extensively studied, and identifying cliques can be very fast by using a variant of the Bron–Kerbosch algorithm [37]. Several variants of this algorithm have been proposed since the original version was published in 1973. We have implemented one of the efficient variants, namely, the one with vertex pivoting and ordering [38]. The parameter k, required by the Bron–Kerborsch algorithm, determines the minimal number of nodes within the cliques to be selected as seeds. When $k = 2$, the cliques reduce to random edges [31]. When $k = 3$ the cliques are equivalent to the triangles.

After enumerating all maximal cliques, the greedy expansion is performed. Given a seed S, which is initialized to be one of the maximal cliques, we obtain its companion set:

$$N = \bigcup_{i \in S} N(i) \tag{18}$$

where i is a node in S and $N(i)$ stands for the neighbors of i. Then, we need to move nodes from N to S to expand S. To this end, we utilize a greedy optimization policy to optimize (16). In our expansion policy, we evaluate the movement of each node i from N to S or from S back to N by subtracting the original objective function value from the new value caused by the movement, i.e.,

$$f_i = f(S \cup \{i\}) - f(S) \quad (i \in N) \tag{19}$$

or

$$f_i = f(S \backslash \{i\}) - f(S) \quad (i \in S). \tag{20}$$

Each node $i \in N$ or S has an f_i value, and we choose the node with the maximal and positive value as

$$j = \arg\max_i f_i \quad (i \in S \cup N, f_i > 0). \tag{21}$$

The movement of this chosen node j is accepted to ensure that the new objective function value exceed the previous one. The movements of all the other nodes are dropped. After the node with the maximal value is selected and moved between N and S, we update N according to S and make sure that N always covers all the neighbors of nodes in S. The same procedure is repeated until no movement of a node between N and S can optimize the fitness function anymore. If the movement of any node i from N to S or from S to N decreases the current objective function value $f(S)$, then the expansion process stops. At this time, we have a set of nodes with the local maximal fitness. This node set is one of the detected overlapping communities. The asymmetric expansion is given in Algorithm 1 in the supplementary material.

However, many different seeds are likely to be expanded into the same overlapping community, which causes the same node set to be detected for multiple times. This duplication problem results in significant running time overhead. In order to save time, we quantify the relative overlap between an expanded node set and all unprocessed seeds using the relative overlap, defined as

$$\sigma_i = \frac{|\mathcal{S}_c \cap \mathcal{S}_u|}{|\mathcal{S}_c \cup \mathcal{S}_u|} \tag{22}$$

where $\mathcal{S}_c$ represents the set of nodes for community c, and $\mathcal{S}_u$ represents the set of nodes for an unprocessed seed. When the relative overlap is high, it is likely that $\mathcal{S}_u$ expands to $\mathcal{S}_c$ again. Therefore, a relative overlap threshold σ is introduced to tell whether seed u will be expanded to c again by checking if $\sigma_i \geq \sigma$.

Note that the relative overlap threshold σ is different from the community distance ϵ in [14]. The community distance evaluates between two communities, and differently σ evaluates between community S_c and an not-yet-processed seed set S_s. Both approaches can reach the same goal, but the relative overlap requires less computing time because the time of comparing a community and a seed is less than that of comparing two communities. In addition, σ is used to judge the excessive similarity between two discovered overlapping communities. As this process is asymmetric by viewing active and static nodes differently, the resulting method is called the asymmetric clique expansion. The work in [39] also distinguished the different types of nodes.

D. ACENV Algorithm

So far, we have covered the details of important components. Next, we present the complete ACENV in Algorithm 2 in the supplementary material. First, we discuss the prerequisite and input data format which are different for static and dynamic networks respectively. Then, its steps are explained, and its time complexity is analyzed. Finally, we discuss the meaning of each parameter as well as the approach of tuning them.

When ACENV is performed on a static network, the input graph g is provided directly to the algorithm, and when it is performed on an evolving network, the input is required to be in the form of sorted collection of network snapshots $\{g_1, g_2, \ldots, g_n\}$ in chronological order, and an index is required to specify which snapshot in the sequence to analyze. Given the minimum clique size k, overlap threshold σ

and parameters α and β, ACENV performs as illustrated in Algorithm 2 in the supplementary material. We first evaluate v_i for each node i, search all maximal cliques to be used as expansion seeds by using the Bron–Kerbosch algorithm [37], and pass parameter k to it. Then, the clique coverage heuristic (CCH) approach is utilized to prevent unnecessary expansion [14]. For each nonexpanded seed, we expand it by moving nodes between N and S to optimize (16). In addition, a stop condition checks if the number of nodes and neighbors of the group reaches 90% of the network. If so, the algorithm stops the expansion immediately. After all the seeds are handled, a final comparison is performed to eliminate duplicate communities; and we preserve communities with the minimum similarity.

As for ACENV's time complexity, there are four main steps: 1) calculating node vitality; 2) finding cliques; 3) duplicating cliques; and 4) performing an expansion process. It is supposed that we have found M communities, n is the size of the largest community, and a node has c neighbors on average. Then, the time complexity of the first step is $O(n)$, and the second step is $O(c^2)$. The time consumed in the third step is almost negligible, because of the Bron–Kerbosch algorithm's efficiency, while most of the time is spent on the last step. To accomplish the expansion for all M communities, we have to consider nc nodes to find the local maximal objective function value for a community. Therefore, the worst case time complexity is $O(n + c^2 + Mnc) = O(Mnc)$, as c is far less than n. Although the complexity seems to be high, the time cost in detecting overlapping communities on large networks can be effectively lowered by setting proper values of parameters α, β, k, and σ. Before applying ACENV on large networks, a good estimation of their density, overlap degree, overlap diversity etc. is necessary.

In most situations, ACENV can complete in a short time because we have adopted CCH optimization and the excessive expansion problem is solved in our presented method.

There are four parameters required by our method, two of which k and α are due to [13] and [14] while β and σ are newly introduced. Although it suffices to use the default setting $k = 4$, one may assign a higher value for a larger or denser network. If one finds too many cliques, higher k values should be used to reduce the number of seeds. The resolution parameter α is generally set as $\alpha = 1$, and it governs the resolution of discovered overlapping communities. To detect larger communities, α should be lower otherwise higher. Parameter β controls the balance between the current snapshot quality and the evolution similarity, which is similar to α in [19]. Larger β is taken more evolution into account in the process of expansion. Parameter σ is a threshold for the percentage of duplicate nodes between groups. A reasonable σ should be used according to the overlap level of network, and a low σ value is set for heavily overlapped networks.

Compared to GCE, we have used a new objective function that is more suitable for an evolving network. By combining the GCE's fitness function and the evolution similarity between present and future, our method not only considers node's connection but also their evolution, such that our method is suitable for evolving networks. The second difference is that we have adopted the compare-and-abandon policy instead of the overlap-and-merge schema in GCE. To explain why the former is better, let us suppose that we have found a group that overlaps with another. We preserve the one with a larger objective function value rather than blindly merging them as done in GCE. Our compare-and-abandon policy provides a good basis for reassignments of overlapping nodes to be described.

IV. Reassigning Overlapping Nodes

Overlapping nodes are very important to the underlying system, and their proper assignment is nontrivial. From a different perspective, the work studied the overlapping nodes [40] in bipartite networks and [41] in weighted complex networks. Xie *et al.* [27] have also found the misassignment problem and suggested to use F-score to evaluate methods' assigning capabilities for overlapping nodes. In order to evaluate F-score, one should evaluate the *precision* which is the number of properly detected overlapping nodes divided by the number of all discovered overlapping nodes, and the *recall* which is the number of properly detected overlapping nodes divided by number of all true overlapping nodes. F-score is defined as

$$\text{F-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{23}$$

Most expansion-based methods, e.g., [20]–[23] fail to detect proper overlapping nodes because they select many nonmember nodes at the early stages of expansion. In order to optimize the fitness function, these improper nodes are added into the group to make W_{out} values be larger. These improper members are viewed as community members, and they are frequently chosen by other communities again, and results in high false positives. According to the analysis of many expansion-based method's output, we have found that although the assignment of overlapping nodes is poor, ACENV reports very close communities to the ground truth overlapping communities. To get rid of these false positives and to detect true overlapping nodes, we introduce an effective reassigning algorithm.

The reassigning algorithm requires a clustering as its input, which can be retrieved by ACENV or other methods. To decide whether a node is a proper overlapping node or not, we introduce the concept of shared community degree for each node in communities. The shared community degree k_i^s of node i is defined to be the number of neighbors which belong to the same communities with node i, and divided by degree k_i of node i, namely,

$$k_i^s = \frac{|N(i) \cap \mathcal{C}(i)|}{k_i} \tag{24}$$

in which $\mathcal{C}(i)$ is the overlapping community containing node i. In many typical communities as characterized by large fitness, we have found that the shared community degrees k_i^s of overlapping nodes are close to those of nonoverlapping members, namely $k_i^s \approx \bar{k}^s$. That is because the degrees k of an overlapping node and a nonoverlapping one are usually in the same range. However, the nodes with obviously large $k_i^s > (1+\xi)\cdot\bar{k}^s$ have a very high probability to be wrongly assigned overlapping nodes. The reason is that their random neighboring

nodes are excessively selected in their corresponding overlapping communities. Here, the parameter $\xi \in (0, 1)$ is used to measure how much fraction to be considered as an obvious excess. But not all nodes with large k_i^s are false positives and the reassignment of one such node usually leads to change other nodes' k_i^s values. Thus we just use them as the candidates in later greedy optimization instead of simply popping them out from their current communities. We also look for nodes with low $k_i^s < (1-\xi)\cdot\bar{k}^s$ as the possibly undiscovered overlapping nodes, and try to push them back to their closest overlapping communities. Using the shared community degree and a greedy selection policy, the reassigning algorithm for reassigning overlapping nodes is given in Algorithm 3 in the supplementary material.

In Algorithm 3 in the supplementary material, we check each group in cover C for possible overlapping nodes. For each node i in candidate set N, we find out all groups containing node i. Next, we try ejecting node i from every group to evaluate new k_i^s values, and choose group c' from which we eject node i to get the minimal $|k_i^s - \bar{k}^s|$ value. When each node is ejected, the shared community degrees of others are recalculated. As long as it does not satisfy $k_i^s > (1+\xi)\cdot\bar{k}^s$, the node is deleted from candidates N in the loop statement.

In a nutshell, the reassigning algorithm for correcting overlapping nodes consists of two steps: 1) excluding improper overlapping nodes and 2) including undiscovered but proper overlapping nodes. Both steps make use of the optimization of shared community degrees, minimizing the difference between an individual's k_i^s and average shared community degree $\bar{k}_s$ over all nodes. Algorithm 3 in the supplementary material can also improve accuracy of overlapping nodes in [42]. The studies [43] and [44] also detect overlapping communities by analyzing the neighbors of a node, and assign nodes to communities by using matrix factorization and finally improved the accuracy. The neighborhood ratio defined in [43] and [44] that is used for building a new adjacency matrix to detect communities, is similar to the shared community degree we discuss here. However, it is dedicated to optimizing the assignment of overlapping nodes.

In addition, $\bar{k}_s$ over all nodes is approximately equal to the average value of the fitness over all discovered communities. This algorithm is able to solve the wrong assignment problem of overlapping nodes and improve the overall accuracy of ACENV to be validated next.

V. Verification and Performance Evaluation

In this section, we verify the effectiveness of our proposed ACENV by comparing it with the state-of-art methods. First, we validate that our objective function effectively solves the excessive expansion problem. Next, we present the performance benchmark results on synthetic networks and a real evolving network to demonstrate its application.

The first step of experiments is to build synthetic networks, which are created by the LFR benchmark. For creating diverse networks, the LFR benchmark provides up to ten parameters which are summarized in Table S2 in the supplementary material. Among these parameters, network size n indicates the scale of the generated networks. It affects the required time to complete community detection, but not the accuracy and predictability of results. The average degree k is set to 15, which is similar to a real-world network's average degree. Parameters τ_1 and τ_2 govern the heterogeneous distribution of degrees and community sizes, respectively. O_n is the number of overlapping nodes, and O_m is the largest number of communities to which a node can belong at present. μ is the fraction of a node's links that connect it to the outside of its community. Our experiments mainly focus on the effects of O_m, O_n, and μ on the results. Note that the other parameters are set by following [14].

A. Objective Function for Excessive Expansion Problem

This experiment aims to verify whether our newly formulated fitness function can help solve the excessive expansion problem. It is performed on the generated LFR network whose parameters are chosen as shown in Table S2 in the supplementary material. A large μ is necessary to produce the excessive expansion problem. First, we generate the LFR networks. Then, we perform GCE (G_1 for short), ACENV with (3) as its objective (A_0 for short), and ACENV with the new fitness function in (16) (A_1 for short) on the generated networks. Finally, we collect sizes of discovered communities, and compare with ground truth (G_0 for short). In order to compare discovered communities' sizes with ground truth, we preserve all the communities in the ground truth and reorder the discovered communities by making the best matched communities aligned with ground truth communities. We use the Jaccard distance similarity to find the best match, and require that one discovered community be matched with only one ground truth community, and the discovered communities with bad matches be dropped. After such ordering and alignment, the results are given in Fig. 1.

In Fig. 1(a), the histogram depicts the sizes of discovered communities of A_0, A_1, G_0, and G_1. Each bar in the histogram corresponds to a community, and the matched communities for G_0, A_0, and A_1 are grouped together. The size of the community detected in A_0 in the first cluster is 1000, however, to save the space, the first bar is truncated. As shown in the histogram, A_0 produces an unreasonable giant community represented by index 1, and the number of matched communities is 9, which is much less than true value 32. It suggests that the excessive expansion problem has definitely appeared. The community sizes of A_1 is much closer to ground truth. There is no giant community, and the number of discovered communities is 27.

In Fig. 1(b), we depict the size difference between discovered and true communities, which is equivalent to the histogram. The result produced by GCE is also included for comparison. Each point stands for a size difference between the matched community and true one, and they are connected by line segments for comparison. The point above 0 indicates that its community size is larger than that of true ones, and point below 0 indicates that its size is smaller.

As seen from Fig. 1, when the excessive expansion problem arises, some communities are not revealed as they are improperly merged into other giant communities, and the number

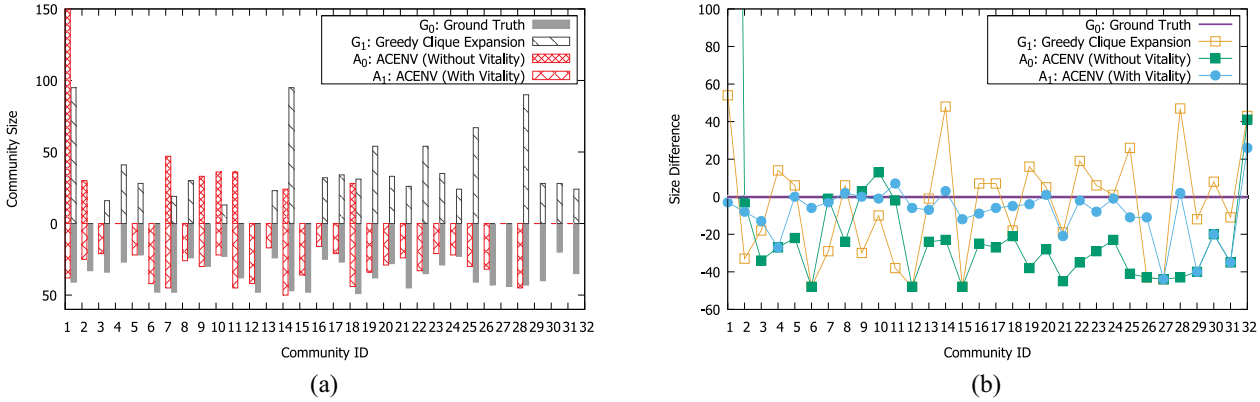


Fig. 1. Sizes of communities and size differences between discovered communities and ground truth communities according to the best-match alignment. Note the bar of ACENV with vitality and ground truth are depicted under the x-axis for space issues.

of communities decreases. In this experiment, A_1 performs much better than A_0 and GCE. Although GCE's excessive expansion problem is not explicitly given here, it is similar to the problem with A_0 in the histogram. GCE's excessive expansion problem indeed exists but its capture is somewhat subtle. The worst case of the excessive expansion problem of GCE is that one community is expanded to the entire network, and at this time, it detects nothing meaningful. These results on the generated network show that the proposed fitness function has improved the performances and made identified communities' sizes closer to ground truth. Therefore, we conclude that it can solve the excessive expansion problem.

B. Performance Benchmark on Synthetic Network

In this experiment, we evaluate ACENV's performance and compare it with the current state-of-art methods, namely GANXiSw [45], COPRA [46], and GCE [47]. We also compare it with the recently proposed neighborhood-inflated seed expansion (NISE) [48]. We generate diverse LFR networks with parameters in Table S2 in the supplementary material, then use these methods and ACENV to them, and finally, adopt NMI and F-score to evaluate the respective results. The experiment is similar to that in Section V-A, but there are more evaluations on the effects of different LFR parameters on the results.

When generating networks, both effects of LFR parameters μ and O_m are investigated, while previous methods have explored only μ or O_m such that they can conceal poor performances brought by the other parameter. To evaluate the detect ability under different levels of connection density inside communities, we create a group of networks as the input by varying μ from 0.1 to 0.8 while fixing O_n to be $10\% \times n$, and choosing O_m from $\{3, 5\}$. For each value of μ, ten networks are generated and the results are averaged for comparison. To evaluate the ability of detecting highly overlapping communities, we create another group of networks by increasing O_m from 2 to 8, setting O_n to be $10\% \times n$ and choosing μ from $\{0.3, 0.5\}$. Higher μ creates decreasing connection densities inside communities, and makes detection harder. Similarly, the task becomes more difficult when O_m becomes larger.

To provide more control when identifying overlapping communities, most methods provide parameters to control the detection process. To perform a fair comparison, we carefully choose their parameters. For COPRA, we set its only parameter v to be equal to O_m. For GANXiSw, we run it multiple times and at each time set its parameter α to different value in $\{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, and keep the result with highest NMI. For GCE, which resembles our method in terms of the expansion strategy, we use the default setting of its parameters, namely $k = 4$ and $\alpha = 1.0$. For NISE, we use the parameter settings with their best performance. For our method, we set $\sigma = 0.75$, $k = 4$, and $\alpha = 1$ for ACENV, and $\xi = 0.3$ for Algorithm 3 in the supplementary material.

After running the compared methods, the output and the ground truth communities generated by the LFR generator are provided to evaluate them. As NMI is widely used to quantify the quality of a clustered result at a community level, we adopt it to evaluate all results. We also adopt F-score as suggested by Xie *et al.* [27], [45] to evaluate the results at the node level. It represents the task of detecting overlapping nodes as a binary classification problem to evaluate the capability of detecting overlapping nodes. Finally, we collect NMIs and F-scores, as shown in Figs. 2 and S2 in the supplementary material, respectively.

In Fig. 2(a), we present the NMIs of GCE, GANXiSw, and COPRA as a function of μ under condition of $O_m = 3$ and $O_n = 10\% \times n$. All four NMIs gradually decrease as μ increases. When $\mu \leq 0.4$, all of the curves exceed 0.6, COPRA's curve is at the bottom, and our method's is at the top. When $\mu > 0.4$, COPRA and GANXiSw decay quickly to 0 when $\mu = 0.6$, while GCE and our method still have high NMIs until $\mu = 0.7$, and decays to 0 when $\mu = 0.8$. Fig. 2(b) is similar to Fig. 2(a), but with higher overlapping diversity $O_m = 5$, and the result suggests that COPRA is slightly unstable. As Figs. 2(a) and (b) show, our method's NMI is nearly the topmost among all the tested methods except for μ is 0.6.

In Fig. 2(c), we compare the detection accuracy of overlapping nodes with these methods as a function of μ under

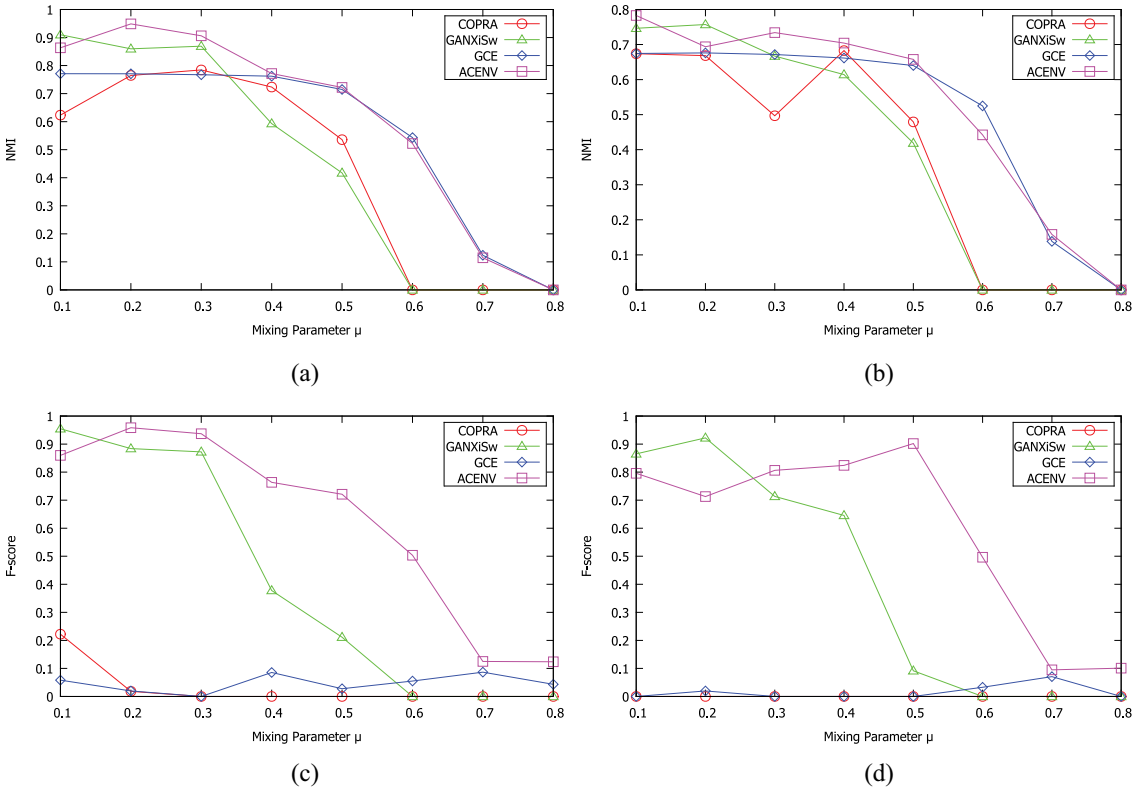


Fig. 2. Performances of COPRA, GANXiSw, GCE, and our method on LFR graphs measured by NMI and F-score as a function of μ. (a) $O_m = 3$ and $O_n = 10\%$. (b) $O_m = 5$ and $O_n = 10\%$. (c) $O_m = 3$ and $O_n = 10\%$. (d) $O_m = 5$ and $O_n = 10\%$.

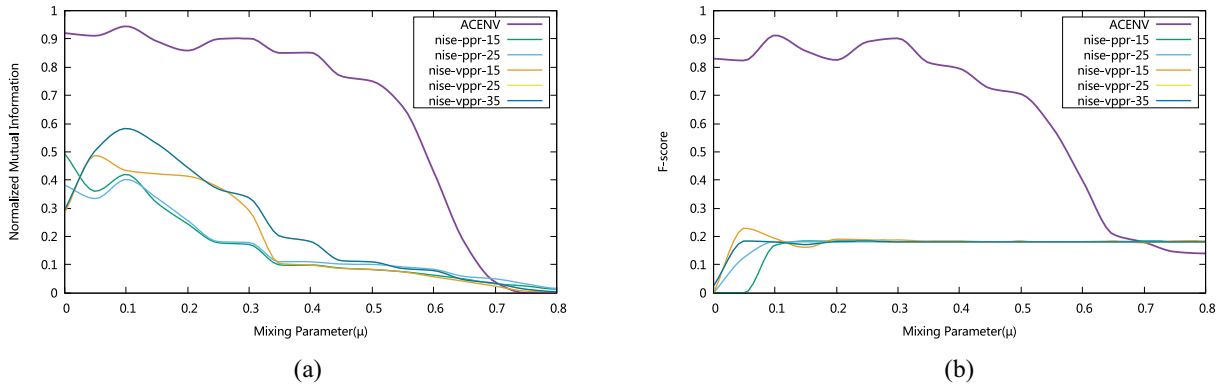


Fig. 3. Performance of ACENV and NISE with $\mu \in (0, 0.8)$ measured by NMI and F-score. (a) $O_n = 10\%$, $O_m = 3$ measured by NMI. (b) $O_n = 10\%$, $O_m = 3$ measured by F-score.

the condition of $O_m = 3$ or $O_m = 5$, and $O_n = 10\% \times n$. It is obvious that COPRA's and GCE's F-scores are almost 0, thereby suggesting that they cannot detect overlapping nodes properly. Although GANXiSw detects overlapping nodes well, our method offers better abilities, as indicated by its F-score. Fig. 2(d) is similar to Fig. 2(c), but under the condition of higher O_m, and the results are similar to Fig. 2(c). Fig. 2(c) and (d) demonstrate that our method improves the performance measured by F-score. This performance improvement at the node level is attributed to the reassigning algorithm to a large extent. However, our method's performance become poor as well when $\mu \geq 0.6$.

NISE is a recently proposed overlapping community detection algorithm [48], and it uses a personalized PageRank clustering scheme to optimize the conductance objective function. To achieve the best performances of NISE, we have run it with personalized PageRank policy (ppr for short) and Fiddler vector personalized PageRank (vppr for short), and tested different values of parameter k, which follow the suggestions in [48]. After performing on the LFR network with $n = 1000, O_n = 100$, and $O_m = 3$, the results comparing with NISE measured by NMI and F-score are given in Fig. 3 and ACENV is better than NISE.

In Fig. S2(a) in the supplementary material, we present how O_m affects the accuracies under $\mu = 0.3$ and $O_n = 10\%$ of n. As expected, all NMIs decay as O_m changes from 1 to 8, and the curve corresponding to our method is the topmost for almost all O_m values. Fig. S2(b) in the supplementary material is similar to Fig. 2(a) except for μ being 0.5. In this case, our method's F-scores exceed others when $O_m \leq 6$, but is

slightly less than GCE when $O_m > 6$. In Fig. S2(c) in the supplementary material, we take a closer look at the F-scores. COPRA's and GCE's F-scores are almost 0, which is consistent to Fig. 2 (c) and (d). When $O_m \leq 5$, ACENV outperforms others. When $O_m > 5$, ACENV is worse than GANXiSw but better than COPRA and GCE. In Fig. S2(d) in the supplementary material, the condition $\mu = 0.5$ is tougher than Fig. 2(c), but our method outperforms others.

The results suggest that the performance measured by NMI and F-score for overlapping nodes demonstrates our method's effectiveness. It is proved that our method has a better ability to detect overlapping communities, and outperforms the current state-of-art methods. At the node level, our method's detection of overlapping nodes is more accurate than other methods. ACENV's output is already very close to ground truth, and the reassigning algorithm makes the results even closer to it. It turns out that ACENV is better than GCE, COPRA, GANXiSw, and the recent NISE under diverse conditions of μ, overlapping degree O_n, and overlapping diversity O_m.

C. Evaluation on Real Network

We demonstrate the application of ACENV to find meaningful communities in real networks by comparing it with other related methods. As there are already many validations in static networks with known ground-truths, we focus our experiments on networks with unknown ground truths. By performing on such networks and measuring the projected clustering for the current snapshot into the next, the ability of ACENV to predict community tendency is also shown. In order to perform the validation, we have built the Fedora package dependency networks as our dataset.

In the Fedora dependency network, a node corresponds to a software package, e.g., glibc, gcc, and gdb. These interdependent packages tend to interact with each other to complete advanced tasks in an operating system, and they are qualified to be meaningful communities. We extract this network from Fedora Linux Distribution Repository,[1] and obtain a sequence of network snapshots.

To construct the networks, we use the package dependency data from repository of Fedora[2] Linux distribution ranging from version 7 to 14. The repository records dependency information that we need, namely what features a package can provide and require. We create a node for each package, and an edge for two nodes if one of them requires any feature which another package provides, and finally obtain a sequence of undirected and unweighted networks. In the process of extraction, we delete the special node standing for glibc, since almost all nodes have edges connecting with it. The statistics of these networks are summarized in Table S3 in the supplementary material, which include the number of nodes, edges, average degree, and maximal degree.

Next, ACENV is performed on each snapshot of the network sequence by using the default parameter settings ($k = 4, \alpha = 1, \beta = 0.1$, and $\sigma = 0.75$), and compare its results with those detected by GCE [14]. We cannot obtain node vitality on the network for version 7 by comparing it with its previous snapshot. Because version 6 is not available, ACENV is performed by viewing it as a static network (by setting $\beta = 0$), and its scores are marked with star in Table S4 in the supplementary material. For a static network, ACENV and GCE are also performed on each snapshot. To compare them, the overlapping modularity [28] for the methods on each snapshot $\Omega(\mathcal{C}_t, g_t)$ is given in Table S4 in the supplementary material. To measure how much development tendency for these communities, we also project the clustering for each snapshot onto the following snapshot and then evaluate overlapping modularity for these projected clustering, namely $\Omega(\mathcal{C}_t, g_{t+1})$. The overlapping modularity is measured between clustering $\mathcal{C}_t$ and current snapshot g_t, while the projected modularity is measured between $\mathcal{C}_t$ and the next snapshot g_{t+1}. It can be seen from Table S4 in the supplementary material that both the overlapping modularity and especially the projected modularity of ACENV exceed GCE's. Table S4 in the supplementary material clearly demonstrates that ACENV not only detects communities well in the current snapshot but also predicts development trends as well.

We have also found many interesting phenomena on the network, for example, after testing ACENV with β ranging from 0 to 1, we have found the results under different β are quite similar. However, to our surprise, by setting $\beta = 1$, Even if we totally ignore the current snapshot, our method can still obtain excellent clustering results. This is because the subsystems in an operating system are usually processed atomically and will not change very much by different versions. By taking a closer look at the results and the package names, we have confirmed the discovered communities. In version 14, for example, ACENV has properly reported communities corresponding to subsystems such as Office, Graphical System, Build System, and Input Method. We have also found that compiling tools, like gcc, gcc-java, elfutils and gcc-c++ are naturally grouped together in the same community, which is true for other snapshots. The detected communities in this evolving networks match most of the known subsystems of Fedora, and this fact proves the effectiveness of ACENV, although the ground truth is unknown. This shows that our proposed ACENV is a useful tool for real-world networks.

VI. Conclusion

The existing methods are limited to performing overlapping community detection in static networks, and fail to handle real-world networks in which nodes are joining or leaving groups, moving from one group to another, and their activeness changes periodically or randomly. This paper presents an overlapping community detection method called ACENV for this type of network. It outperforms other state-of-the-art methods under the diverse conditions of overlaps by NMI and F-score. The main contributions of this paper include:

1) A new concept called node vitality is introduced to quantify each node's activeness in evolving networks, and the method of obtaining its values by comparing consecutive network snapshots is presented.

[1] http://archives.fedoraproject.org/

[2] http://fedoraproject.org/

2) A method of embedding dynamic information into a fitness function is presented.
3) Based on node vitality and improved fitness function, an expansion-based overlapping community detection algorithm is proposed.
4) Finally, a reassigning algorithm for overlapping nodes to improve the results obtained by ACENV comparing with other overlapping community detection methods is given.

The future work should consider the use of advanced learning and optimization methods, e.g., [49]–[55], to achieve higher overlapping community detection accuracy.

REFERENCES

[1] L. Liu and H. Jia, "Trust evaluation via large-scale complex service-oriented online social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 11, pp. 1402–1412, Nov. 2015.

[2] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[3] J. Eustace, X. Wang, and J. Li, "Approximating Web communities using subspace decomposition," *Knowl. Based Syst.*, vol. 70, pp. 118–127, Nov. 2014.

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

[5] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

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

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Disc. Data Min.*, vol. 96. Portland, OR, USA, 1996, pp. 226–231.

[8] 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.

[9] J. Li, X. Wang, and Y. Cui, "Uncovering the overlapping community structure of complex networks by maximal cliques," *Physica A Stat. Mech. Appl.*, vol. 415, pp. 398–406, Dec. 2014.

[10] D. Defays, "An efficient algorithm for a complete link method," *Comput. J.*, vol. 20, no. 4, pp. 364–366, 1977.

[11] E. L. Martelot and C. Hankin, "Fast multi-scale detection of overlapping communities using local criteria," *Computing*, vol. 96, no. 11, pp. 1011–1027, 2014.

[12] J. M. Hofman and C. H. Wiggins, "Bayesian approach to network modularity," *Phys. Rev. Lett.*, vol. 100, no. 25, 2008, Art. no. 258701.

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

[14] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," in *Proc. 4th Int. Workshop Soc. Netw. Min. Anal.*, Washington, DC, USA, 2010, pp. 33–42.

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

[16] J. Cheng *et al.*, "Routing in Internet of Vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2339–2352, Oct. 2015.

[17] Y.-Q. Zhang, X. Li, J. Xu, and A. V. Vasilakos, "Human interactive patterns in temporal networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 214–222, Feb. 2015.

[18] C. Wang, L. Cao, and C.-H. Chi, "Formalization and verification of group behavior interactions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 8, pp. 1109–1124, Aug. 2015.

[19] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Philadelphia, PA, USA, 2006, pp. 554–560.

[20] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2007, pp. 153–162.

[21] M.-S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 622–633, 2009.

[22] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: A framework for analyzing communities and their evolutions in dynamic networks," in *Proc. 17th Int. Conf. World Wide Web*, Beijing, China, 2008, pp. 685–694.

[23] K. Liu *et al.*, "Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks," *Knowl. Based Syst.*, vol. 89, pp. 487–496, Nov. 2015.

[24] J. Li, K. Yu, and K. Hu, "A novel dynamical community detection algorithm based on weighting scheme," *Int. J. Modern Phys. C*, vol. 26, no. 8, 2015, Art. no. 1550091.

[25] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné, "Trust and compactness in social network groups," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 205–216, Feb. 2015.

[26] G. Ren and X. Wang, "Epidemic spreading in time-varying community networks," *Chaos Interdiscipl. J. Nonlin. Sci.*, vol. 24, no. 2, 2014, Art. no. 023116.

[27] 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, p. 43, 2013.

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

[29] G. Bianconi and A.-L. Barabási, "Competition and multiscaling in evolving networks," *Europhys. Lett.*, vol. 54, no. 4, p. 436, 2001.

[30] J. P. Bagrow and E. M. Bollt, "Local method for detecting communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 4, 2005, Art. no. 046108.

[31] 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. IADIS AC*, vol. 5, 2005, pp. 97–104.

[32] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, 2004, Art. no. 066111.

[33] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, 2009, Art. no. 016118.

[34] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 1, 2011, Art. no. 016107.

[35] Y. Cui, X. Wang, and J. Eustace, "Detecting community structure via the maximal sub-graphs and belonging degrees in complex networks," *Physica A Stat. Mech. Appl.*, vol. 416, pp. 198–207, Dec. 2014.

[36] Y. Cui, X. Wang, and J. Li, "Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient," *Physica A Stat. Mech. Appl.*, vol. 405, pp. 85–91, Jul. 2014.

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

[38] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in sparse graphs in near-optimal time," in *Proc. Int. Symp. Algorithms Comput.*, 2010, pp. 403–414.

[39] X. Wang and X. Qin, "Asymmetric intimacy and algorithm for detecting communities in bipartite networks," *Physica A Stat. Mech. Appl.*, vol. 462, pp. 569–578, Nov. 2016.

[40] Y. Cui and X. Wang, "Uncovering overlapping community structures by the key bi-community and intimate degree in bipartite networks," *Physica A Stat. Mech. Appl.*, vol. 407, pp. 7–14, Aug. 2014.

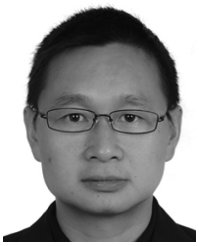
[41] J. Li, X. Wang, and J. Eustace, "Detecting overlapping communities by seed community in weighted complex networks," *Physica A Stat. Mech. Appl.*, vol. 392, no. 23, pp. 6125–6134, 2013.

[42] X. Wang and J. Li, "Detecting communities by the core-vertex and intimate degree in complex networks," *Physica A Stat. Mech. Appl.*, vol. 392, no. 10, pp. 2555–2563, 2013.

[43] J. Eustace, X. Wang, and Y. Cui, "Overlapping community detection using neighborhood ratio matrix," *Physica A Stat. Mech. Appl.*, vol. 421, pp. 510–521, Mar. 2015.

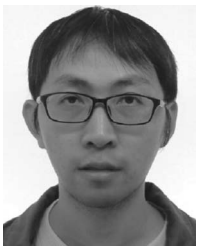
[44] J. Eustace, X. Wang, and Y. Cui, "Community detection using local neighborhood in complex networks," *Physica A Stat. Mech. Appl.*, vol. 436, pp. 665–677, Oct. 2015.

[45] 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 Min. Workshops (ICDMW)*, Vancouver, BC, Canada, 2011, pp. 344–349.
[46] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, 2010, Art. no. 103018.
[47] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," in *Proc. Int. Conf. Adv. Soc. Netw. Anal. Min. (ASONAM)*, 2010, pp. 176–183.
[48] 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.
[49] L. Li, Y. Lin, N. Zheng, and F.-Y. Wang, "Parallel learning: A perspective and a framework," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 3, pp. 389–395, 2017.
[50] X. Luo *et al.*, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.
[51] Z.-H. You, M. C. Zhou, X. Luo, and S. Li, "Highly efficient framework for predicting interactions between proteins," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 731–743, Mar. 2017.
[52] W. Dong and M. C. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1135–1148, Jul. 2017.
[53] W. Gu, Y. Yu, and W. Hu, "Artificial bee colony algorithmbased parameter estimation of fractional-order chaotic system with time delay," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 107–113, Jan. 2017.
[54] W. Han *et al.*, "Cuckoo search and particle filter-based inversing approach to estimating defects via magnetic flux leakage signals," *IEEE Trans. Mag.*, vol. 52, no. 4, Apr. 2016, Art. no. 6200511.
[55] Q. Kang, S. W. Feng, M. C. Zhou, A. C. Ammari, and K. Sedraoui, "Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2557–2568, Sep. 2017.



Jiujun Cheng received the Ph.D. degree in computer application technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006.

He is currently a Professor with Tongji University, Shanghai, China. In 2009, he was a Visiting Professor with Aalto University, Espoo, Finland. He has over 50 publications including conference and journal papers. His current research interests include mobile computing, complex networks with a focus on mobile/Internet interworking, service computing, and Internet of Vehicles.



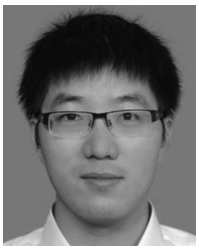
Xiao Wu received the B.S. degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010. He is currently pursuing the master's degree with the College of Electronic and Information Engineering, Tongji University, Shanghai, China.

His current research interest includes complex networks.

Mengchu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of Electrical and Computer Engineering. He has over 700 publications, including 12 books, over 400 journal papers (over 290 in IEEE TRANSACTIONS), 11 patents, and 28 book-chapters. His current research interests include Petri nets, intelligent automation, Internet of Things, big data, Web services, and intelligent transportation.

Dr. Zhou was a recipient of the Humboldt Research Award for U.S. Senior Scientists from Alexander von Humboldt Foundation, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from IEEE Systems, Man, and Cybernetics Society. He is the Founding Editor of IEEE Press Book Series on Systems Science and Engineering. He is a Life Member of the Chinese Association for Science and Technology—USA and served as its President in 1999. He is a Fellow of the International Federation of Automatic Control, the American Association for the Advancement of Science, and Chinese Association of Automation.



Shangce Gao (M'11–SM'16) received the Ph.D. degree in innovative life science from the University of Toyama, Toyama, Japan, in 2011.

He is currently an Associate Professor with the University of Toyama. He has authored over 90 publications in journals and conference proceedings. His current research interests include mobile computing, machine learning, and neural networks.

Dr. Gao was a recipient of the Best Paper Award at the IEEE 2016 International Conference on Progress in Informatics and Computing, the Shanghai Rising-Star Scientist Award, the Chen-Guang Scholar of Shanghai Award, the Outstanding Academic Performance Award of The Institute of Electronics, Information and Communication Engineers, and the Outstanding Academic Achievement Award of Information Processing Society of Japan.

Zhenhua Huang received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2008.

He is currently a Professor with the School of Electronics and Information, Tongji University, Shanghai. He has published over 50 papers in various journals and conference proceedings. His current research interests include data warehouse, online analytical processing applications, data mining, and knowledge discovery.

Cong Liu received the B.S. and M.S. degrees in computer software and theory from the Shandong University of Science and Technology, Qingdao, China, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Section of Information Systems, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

His current research interests include process mining, Petri nets, and software process mining.