Clustering dynamics of complex discrete-time networks and its application in community detection
Jianshe Wu and Yang Jiao

Citation: Chaos: An Interdisciplinary Journal of Nonlinear Science 24, 033104 (2014); doi: 10.1063/1.4886695
View online: http://dx.doi.org/10.1063/1.4886695
View Table of Contents: http://scitation.aip.org/content/aip/journal/chaos/24/3?ver=pdfcov
Published by the AIP Publishing

Articles you may be interested in
Stability analysis and synchronization in discrete-time complex networks with delayed coupling
Chaos 23, 043108 (2013); 10.1063/1.4825095

Global stability analysis of discrete-time coupled systems on networks and its applications
Chaos 22, 033135 (2012); 10.1063/1.4748851

Outer synchronization of coupled discrete-time networks
Chaos 19, 013106 (2009); 10.1063/1.3068357

Discretetime modeling of atmospheric turbulence with application to sonic booms
J. Acoust. Soc. Am. 120, 3078 (2006); 10.1121/1.4787414

Discretetime detection modeling for unsaturated ocean acoustic propagation
J. Acoust. Soc. Am. 74, 1630 (1983); 10.1121/1.390126
Clustering dynamics of complex discrete-time networks and its application in community detection

Jianshe Wu and Yang Jiao
The Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xi’an Jiaotong University, Xi’an 710071, China

(Received 23 April 2014; accepted 23 June 2014; published online 3 July 2014)

The clustering phenomenon is common in real world networks. A discrete-time network model is proposed firstly in this paper, and then the phase clustering dynamics of the networks are studied carefully. The proposed model acts as a bridge between the dynamic phenomenon and the topology of a modular network. On one hand, phase clustering phenomenon will occur for a modular network by the proposed model; on the other hand, the communities can be identified from the clustering phenomenon. Beyond the phases’ information, it is found that the frequencies of phases can be applied to community detection also with the proposed model. In specific, communities are identified from the information of phases and their frequencies of the nodes. Detailed algorithm for community detection is provided. Experiments show that the performance and efficiency of the dynamics based algorithm are competitive with recent modularity based algorithms in large scale networks. © 2014 AIP Publishing LLC. [http://dx.doi.org/10.1063/1.4886695]

The clustering phenomenon is common in real world networks but seldom reported in literatures. In this paper, the classical Kuromoto model is modified as a discrete-time network model, which exhibits clustering phenomenon. The clustering dynamics are analyzed in detail. Community detection is also a challengeable problem in complex networks. Traditional methods for community detection usually use a structure based measure, e.g., the modularity, or a function based measure, e.g., the spectral cohesion. Since the clustering phenomenon bridges the gap between the structure and function of networks, we use the phases and their frequencies of the clustering phenomenon to detect communities, detailed algorithm is provided. The performance and efficiency of the proposed discrete-time dynamics (DTD) algorithm are tested by comparative experiments on large scale networks with 50 000 nodes.

I. INTRODUCTION

The dynamics of networks have been a research focus in the society of complex networks, especially the synchronization, percolation, and epidemics phenomena. Many real world networks exhibit modular structures, e.g., the social networks, economics networks, biological information networks, and so on. A modular network is composed of several groups or clusters, nodes connect densely with each other in the same community and sparsely in different communities. Besides the network dynamics, finding the community structure, known as community detection, is another challengeable problem in networked systems. Various methods have been proposed, including the structure based methods (e.g., the modularity optimization) and the function based methods (e.g., the spectral cohesion).

In fact, the structure and function (dynamics) in real world networks are closely related. So unveiling the relations between the modular structures and the functions (dynamics or behavioral characteristics) is of more significance, which includes two aspects simultaneously: (i) predicate or evaluate the dynamics of the modular networks and (ii) identify the communities from its dynamic behaviors. In fact, in a social network, a person identifies the communities from the individual’s behaviors (dynamics) other than a graph or structure.

In aspect (i), the dynamics of modular networks have been carefully studied in recent years, most of them focused on the synchronization phenomenon (including cluster synchronization) and synchronization interfaces were observed. In aspect (ii), some pioneer works explored the methods to reveal the community structure via synchronization. The basic mechanism is that the nodes in the same community synchronize before the nodes in different communities. In other words, they use the transient phase clustering process towards global synchronization.

Because the phase clustering phenomenon toward global synchronization is transient, comparing with the phase synchronization, stable phase clustering is more appropriate to bridge the gap between the modular structure and the network function. In Ref. 37, it is recommended to use stable clustering phenomenon for community detection. The stable phases of the nodes in the same community are almost synchronized, whereas the states of the nodes in different communities are far from synchronization. But with the original algorithm stated in Ref. 37, which uses only phases information for community detection, a network with many (more than three usually) communities is not applicable because the stable phases of two or more communities may be mixed together. The methods either by phase synchronization or phase clustering in previous works use an oscillator model governed by differential equation. It is well known...
that the computation of differential equations is much time-consuming than difference equations.

In this paper, a discrete-time network clustering model governed by a difference equation is provided, which is a modification of the famous continuous-time Kuramoto phase oscillator model.\textsuperscript{38} Similar as that in Ref. 37, the phases of two or more communities may be mixed when the network has many communities, but the frequencies of the phases of different communities are different. Thus they can be identified by the frequencies. Based on the discrete-time clustering model, an algorithm for community detection is proposed. By using both the phases information and the frequencies information, the proposed DTD algorithm is competitive with recent structure (modularity) based algorithms in experiments (see Sec. IV, the number of nodes is 50 000).

This paper is organized as follows. The proposed discrete-time model is described in Sec. II, its dynamics are also analyzed. The DTD algorithm is described in Sec. III. Experimental results are demonstrated in Sec. IV. Section V concludes this work.

II. THE DISCRETE-TIME NETWORK AND ITS CLUSTERING DYNAMICS

The clustering phenomenon has been observed in continuous-time networks with modular structure.\textsuperscript{37} The stable phases of the nodes are clustered, in other words, the distance of stable phases between two nodes in the same community is short and the distance between two nodes in different communities is relatively long. The interesting phenomenon was tried to apply in community detection by using the information of stable phases. When the number of communities is more than 3, the phases of different communities may mix together, thus are difficult to be separated by the method proposed in Ref. 37. The following model is proposed in this paper.

A. Model of the discrete-time networks

The coupled Kuramoto oscillators model is described by\textsuperscript{38}

\[
\frac{d\theta_i}{dt} = \omega_i + \frac{d}{k_i} \sum_{j=1}^{N} a_{ij} \sin(\theta_j - \theta_i), i = 1, \ldots, N, \tag{1}
\]

where \(\theta_i\) and \(\omega_i\) are the phase variable and intrinsic frequency of the \(i\)th oscillator, respectively, \(k_i\) is the degree of the \(i\)th oscillator, \(d\) is the coupling strength, \(N\) indicates the number of nodes, and \([a_{ij}]\) is the adjacency matrix of the network.

In this paper, the proposed model is described by the following difference equation:

\[
\theta_i(k+1) = \theta_i(k) + \frac{K}{k_{\text{max}}} \sum_{j=1}^{N} a_{ij} \sin(\theta_j(k) - \theta_i(k)), i = 1, \ldots, N,
\]

\[
\begin{cases}
K = K_p > 0, & e_{ij} \in E, \\
K = K_n < 0, & e_{ij} \notin E,
\end{cases}
\tag{2}
\]

where \(k_{\text{max}}\) indicates the maximum degree of the nodes, which can be replaced by \(N\) in (2); \(K\) is the coupling strength between nodes, \(E\) is the set of edges, and \(\bar{E}\) is the complement set of \(E\). In other words, if an edge \(e_{ij} \in E\), then \(e_{ij} \notin \bar{E}\); if otherwise \(e_{ij} \notin E\), then \(e_{ij} \in \bar{E}\). The initial phase \(\theta_i(0)\) is randomly and uniformly distributed in \([0, 2\pi]\).

When there is a connection between nodes \(i\) and \(j\) \((e_{ij} \in E), K = K_p\) is positive, which will make the phases of two nodes evolve together. When no connection exists between nodes \(i\) and \(j\) \((e_{ij} \notin E), K = K_n\) is negative, which will make the phase of two unconnected nodes evolve far away.

Comparing (2) with (1), the modification includes two points: (i) the differential equation is replaced by difference equation, which makes the algorithm much fast and can run on large scale networks; (ii) negative coupling strength is introduced when there is no connection, which makes a stable phase clustering phenomenon. Negative coupling strength is common in real world networks.\textsuperscript{30,39,40}

B. Dynamics of the discrete-time networks

The dynamic behaviors of model (2) are exhibited by three example networks. The first network is shown in Fig. 1(a), which is a very simple 14-node network with two communities \(C_1\) and \(C_2\): \(C_1 = \{1, 2, 3, 4, 5, 6, 7, 9, 12\}\) and \(C_2 = \{8, 10, 11, 13, 14\}\). Fig. 1(b) is the phases evolving by three example networks. The first network is shown in Fig. 1(a), which is a very simple 14-node network with two communities \(C_1\) and \(C_2\): \(C_1 = \{1, 2, 3, 4, 5, 6, 7, 9, 12\}\) and \(C_2 = \{8, 10, 11, 13, 14\}\). Fig. 1(b) is the phases evolving with iterations by (2) when \(K_p = 2\) and \(K_n = -1\). The initial phases of the 14 nodes are uniformly and randomly distributed in \([0, 2\pi]\). The phases of nodes are evolving into two clusters gradually as the iteration. The two communities can be easily separated by their phases except node 2. Node 2 has four links with \(C_1\) and three links with \(C_2\), it acts as a bridge between the two communities, thus its phase also bridges between the phases of the two communities.

The second and third networks are generated from the Lancichinetti, Fortunato, and Radicchi’s (LFR) benchmark.\textsuperscript{41} The LFR benchmark is widely accepted to generate realistic modular networks for community detection.\textsuperscript{19} In the LFR benchmark, both the node degree and community size are power law distributions, with exponents \(\gamma\) and \(\beta\),
respectively. \( N \) is the number of nodes, \( \langle k \rangle \) is the average degree, and \( k_{\text{max}} \) is the maximum degree. Another important parameter is the mixing parameter \( \mu \), which means each node shares a fraction \( 1 - \mu \) of its connections with the other nodes of its community and a fraction \( \mu \) with the other nodes of the network.

The second network is with parameters \( N = 100, \langle k \rangle = 15, k_{\text{max}} = 30, \gamma = 2, \beta = 1, \) and \( \mu = 0.1 \), and the third network is with parameters \( N = 300, \langle k \rangle = 30, k_{\text{max}} = 60, \gamma = 2, \beta = 1, \) and \( \mu = 0.1 \). The second network has five communities from both the phase’s information and the frequencies information of the curves, thus overcomes the mixing of collective dynamic behavior made it possible to detect the frequencies of the phase curves in Fig. 2. The new finding communities can be visually identified easily, the reason is that the nodes upset dramatically, they do not change as a whole. But the frequencies information can be applied to community detection.

In previous works of detecting communities from network dynamics, only the phase information in specific time was used. In both Figs. 2(a) and 2(b), the communities cannot be always identified correctly if only the phase information of the nodes is used, because the stable phases of several communities are mixed at specific number of iterations. For example, in Fig. 2(a), \( C_1 \) and \( C_2 \) are mixed in the iteration number from about 18 to 30; in Fig. 2(b), \( C_4 \) and \( C_5 \) are mixed in the iteration number from about 85 to 146. But the communities can be visually identified easily, the reason is that the frequencies of the phase curves in Fig. 2. The new finding of collective dynamic behavior made it possible to detect communities from both the phase’s information and the frequencies information of the curves, thus overcomes the shortcoming in Ref. 37 and is applicable in networks with many communities.

C. The slowly decaying state

Lee et al. have studied the frequency clustering in one-dimensional chain and two-dimensional lattice of coupled phase oscillators.\(^{42,43}\) As shown in Fig. 2, from the randomly generated initial phases, after a period of transient oscillatory process, the phases of the nodes are evolving slowly. In this slowly decaying state, the phases of the nodes are not static absolutely. Instead, the phases of nodes in the same community change slowly as a whole, thus the communities can be detected. In the transient oscillatory process, the phases of the nodes upset dramatically, they do not change as a whole. Thus, the frequencies of the phases iteration can be a criterion for judging whether it is ready for community detection.

The frequency of the phase curve of node \( i \) at the \( k \)th iteration is calculated as follows:

\[
 f_i(k) = \frac{\theta_i(k) - \theta_i(k - \text{len})}{\text{len}} ,
\]

where \( \text{len} \) is the iteration length to calculate the frequency. Obviously, the frequency calculated by (3) is the average frequency within the iteration length \( \text{len} \). Given a threshold \( \xi \), if \( f_i(k) < \xi \), then the network is judged as in the slowly decaying state.

D. Mathematical analysis

Due to the positive couplings, the phases in the same communities evolve together gradually; due to the negative couplings, the phases in different communities evolve far away. In the slowly decaying state, the frequencies of the nodes in the same community are almost equal but are quite different in different communities; the phases in the same community are close together but the phases of two or more communities may mixed together. Both the phases information and the frequencies information can be applied to community detection.

Equation (2) is difficult to analyze directly. For simplicity to analyze, we assume

\[
\begin{align*}
\sin(\theta_j(k) - \theta_i(k)) &= \theta_j(k) - \theta_i(k), \\
0 &\leq \theta_j(k) - \theta_i(k) < \pi/2; \\
\sin(\theta_j(k) - \theta_i(k)) &= \pi - \theta_j(k) + \theta_i(k), \\
\pi/2 &\leq \theta_j(k) - \theta_i(k) < 3\pi/2; \\
\sin(\theta_j(k) - \theta_i(k)) &= \theta_j(k) - \theta_i(k) - 2\pi, \\
3\pi/2 &\leq \theta_j(k) - \theta_i(k) < 2\pi.
\end{align*}
\]

In Eq. (4), the sin function is approximated by line segments. With this approximation, Eq. (2) becomes

\[
\theta(k + 1) = T\theta(k) + c(k),
\]

where \( T \) is the transfer matrix, \( \theta(k) = (\theta_1(k), \theta_2(k), \ldots, \theta_N(k))^T \) is the phase vector, \( c(k) = (c_1(k), c_2(k), \ldots, c_N(k))^T \) is an \( N \) dimensional vector with components \( c_i(k) = 0, \pi, \) or \(-2\pi \) for \( i = 1, \ldots, N \) (see (4)). From (2) and (4), the row sum of \( T \) is \( 1 + \sum_{j=1}^{N} -\sum_{j=1}^{N} = 1 \), indicating \( \lambda_1 = 1 \) is always an eigenvalue of \( T \). Obviously, \( T \) is time-varying with the iteration. It is not sure that \( |\lambda_i| \leq 1 \)
always holds, but after a period of iteration, \( \lambda_1 = 1 \) will become the largest eigenvalue, the rest eigenvalues correspond to the decaying modes, and the network will converge to the stable state corresponding to the largest eigenvector. In this case, the eigenvalues are denoted as \( \lambda_1 = 1 > \lambda_2 \geq \cdots \geq \lambda_N > -1 \). In general cases before 1 becomes the largest eigenvalue, the eigenvalues are denoted as \( \lambda_1 = 1 \) and \( \lambda_2 \geq \cdots \geq \lambda_N \). The modes corresponding from \( \lambda_2 \) to \( \lambda_N \) are also called decaying modes in this paper.

If \( c_i(k) = 0 \) or \( -2\pi \) for all \( i = 1, \cdots, N \) and \( k \) \( \equiv 0 \mod 2\pi \) equals to \( 0 \) when \( \mod 2\pi \) is applied in the computation, Eq. (5) becomes

\[
\theta(k + 1) = T \theta(k).
\]  

Equation (6) always has an eigenvector \((1, \cdots, 1)^T\) corresponding to the eigenvalue value \( \lambda = 1 \), indicating that the network will be decayed to the global synchronization state in which all nodes have the same value of phases. In this extreme case, the communities can only be detected from the largest decaying eigenvectors.

Ref. 44 provides a method to detect community from diffusion of random walkers, where the diffusion equation is \( \rho(k + 1) = T \rho_k \), which is similar to (6) of this paper. In Ref. 44, \( \rho_i \) is the number of walker at node \( i \). They detect communities by the slowest decaying eigenvectors of the transfer matrix corresponding to the currents in the edges. In this paper, the slowest decaying eigenvectors correspond to the frequencies of the phases.

The global synchronization of the network can be achieved by let \( K_p = 0 \) and large enough \( K_p \). In general, \( c(k) \neq 0 \) for some \( i \) in \( c(k) \) of (5), instead of global synchronization, phases clustering will be the stable state corresponding to \( \lambda_1 = 1 \) after the slowly decaying state.

Suppose \( \tau_2 \geq \tau_3 \geq \cdots \geq \tau_N \) is the decaying time (number of iterations required) for the modes corresponding to \( \lambda_2, \cdots, \lambda_N \), respectively, all the \( N-1 \) modes are converged to the stable phase clustering state corresponding to \( \lambda = 1 \) after \( k > \tau_2 \) in the iteration. If \( \tau_2 > k > \tau_3 \), then the \( N-2 \) modes corresponding to \( \tau_1 \geq \tau_2 \geq \cdots \geq \tau_N \) are converged, but the slowest mode corresponding to \( \lambda_2 \) is still decaying. The frequencies obtained from (3) are the largest decaying eigenvector of the transfer matrix. That is

\[
f_i(k) = b e_i^{(2)}(k),
\]

where \( e_i^{(2)}(k) = (e_i^{(2)}(k), \cdots, e_i^{(2)}(k))^T \) is the largest decaying eigenvector of \( T \) corresponding to \( \lambda_2 \) and \( b \) is a constant.

Fig. 3 is the simulation results obtained from the 100-node LFR network (see Fig. 2a). Fig. 3(a) shows the relation between the frequencies obtained by (3) at \( k = 200 \) and the largest eigenvector \( e^{(2)}(k) \), which coincides with (7) indicating that only the slowest mode is decaying at \( k = 200 \).

If \( \tau_3 > k > \tau_4 \), then there are two slowest modes corresponding to \( \lambda_2 \) and \( \lambda_3 \) are still decaying. The frequencies obtained from (3) are the weighted average of the two largest decaying eigenvectors of the transfer matrix. That is

\[
f_i(k) = b(w_1 e_i^{(2)}(k) + w_2 e_i^{(3)}(k)), \quad w_1 + w_2 = 1,
\]

where \( e_i^{(3)}(k) = (e_i^{(3)}(k), \cdots, e_i^{(3)}(k))^T \) is the second largest decaying eigenvector of \( T \) corresponding to \( \lambda_3 \) and \( w_1 \) and \( w_2 \) are weights of the two eigenvectors. Fig. 3(b) shows the relation between the frequencies obtained by (3) at \( k = 50 \) and the weighted average of the two largest eigenvectors: \( 0.66e^{(2)} + 0.34e^{(3)} \), which coincides with (8) indicating that there are two slowest modes decaying at \( k = 50 \). In general, the frequencies obtained by (3) are weighted average of several largest eigenvectors of the transfer matrix.

In fact, using only the frequencies obtained by (3) at different numbers of iteration, the communities can be also detected. The principle is similar to the spectrum clustering methods which use the largest eigenvectors. Fig. 4 shows the distribution of the frequencies obtained by (3) when \( k = 50 \) and \( k = 200 \), in which the five communities can be clearly identified. Different from the method in Ref. 44, where the eigenvalues and eigenvectors have to be computed, the time complexity will be at least \( O(N^3) \). In this
paper, the (average) frequencies can be computed directly by (3). In addition to the frequencies, the phases information are also applied to community detection.

E. Conditions for the clustering phenomenon

There are two critical conditions for the clustering phenomenon, which is used for community detection. The first condition is that the evolving frequencies of the nodes in the same community are almost equal and the phases of the nodes in the same community are close together. In other words, the phases of nodes in the same community evolving as a whole. The components of the slowest decaying eigenvalues corresponding to the nodes of the same community are almost identical. This condition can be ensured by the lowest threshold of $K_p$, the positive couplings among nodes in the community.

The second condition is that the phases of nodes in any two different communities do not always evolving as a whole (two communities merged as one community). This condition can be ensured by the lowest threshold of $|K_n|$, the negative couplings between nodes in different communities. Breaking of the conditions, e.g., if $K_n = 0$, the network may be globally synchronized with a larger value of $K_p$, the entire network evolves as a whole, thus no community can be detected from the phases information.

The effects of the positive coupling and the negative coupling are opposite, they should be balanced for the slowly decaying state. That is the ratio of $R = K_p/|K_n|$ should be in a range. Too large of $R$ tends to the effect positive coupling and too small of $R$ tends to the effect of negative coupling. Though mathematically obtaining the lowest thresholds of $K_p$ and $|K_n|$ and the range of $R$ is difficult, in a wide range of the parameters, the communities can be detected from the slowly decaying state. In Sec. III, discussion about how to select the values of $K_p$ and $K_n$ will be given.

III. COMMUNITY DETECTION BY CLUSTERING DYNAMICS

With the proposed discrete-time network model, the DTD algorithm is proposed for community detection from both the phases in a specific number of iteration and the average frequencies of the phases.

A. Detect communities from phases

As shown in Fig. 2(a), after about 10 times of iteration, the phases of the nodes are almost stable (decaying slowly). The phases are distributed in $[0, 2\pi]$. The phases of the nodes in the same community are close together and the phases of the nodes in different communities are far away relatively. Divide $[0, 2\pi]$ into $num_p$ number of sub-intervals and calculate the number of nodes whose phases fall into each sub-interval, then the phase distribution curve is obtained. Fig. 5(a) is the phase distribution curve obtained from data of Fig. 2(a) when $k = 24$ (after 24 times of iteration), where $num_p = 40$. It is seen that the five-community network is identified as four communities, communities $C_1$ and $C_2$ are identified as one community because their phases are mixed together. In previous work of identifying community from phase clustering of continuous-time network, the phase mixture phenomenon of two or more communities made it unapplicable to networks with more than three communities. In this paper, the mixed communities can be further identified by the frequencies of phases evolving, which will be introduced in Sec. III B.

B. Detect communities from frequencies

From Fig. 2(a), community $C_1$ and $C_2$ can be clearly identified from the frequencies of their phases evolving, though they cannot be identified by the phases when $18 \leq k \leq 30$.

For a community identified by the phases, calculate the frequency of each node by (3). Divide the frequency interval into several (denoted as $num$, in the context) sub-intervals. Calculate the number of nodes whose frequencies fall into each sub-interval, then the frequency distribution curve is obtained. Fig. 5(b) is an example of the frequency distribution curve obtained from the mixed community $C_1 + C_2$ in Fig. 2(a). The nodes in the same community have similar frequencies, whereas frequencies of nodes in different communities differ greatly. Obviously, $C_1$ and $C_2$ can be easily separated by the frequency distribution curve.

Each of the communities identified by the phases will be further checked by the frequencies to decide whether it is a community or a mixture of two or more communities. Detailed algorithm is described in Sec. III C.

C. Algorithm from clustering dynamics

The steps of the DTD algorithm are as follows:

Step 1: For a given network, generate its adjacency matrix.
Step 2: Randomly generate the initial phases $0_i(0)$ for the nodes.
Step 3: Iteration.

Iteratively compute the phases of the nodes by (2). Examine the condition to stop iteration: $S_i(k) < \zeta$, for all nodes. If the condition is not satisfied, repeat for Step 3, till the condition holds. $\zeta$ is the threshold to judge whether the phases are
evolving slowly. In all the simulations of this paper, \( \xi = 0.1 \) and \( \text{len} = 3 \).

**Step 4:** Identify communities by phases.
Obtain the distribution curve of the phases of all nodes in \([0, 2\pi]\). Divide the interval \([0, 2\pi]\) into \( n_{\text{p}} \) sub-intervals. Count the number of the nodes in each sub-interval and draw the distribution curve of the phases of all nodes in \([0, 2\pi]\), as is given in Fig. 5(a). Nodes between two valleys in the distribution curve are assigned in the same community, and the number of communities is the number of peaks in the distribution curve. The 100-node network shown in Fig. 2(a) is divided into four communities by the dotted line in Fig. 5(a) when \( k = 24 \).

**Step 5:** Small communities reassignment.
Some of the communities obtained from Step 4 may have only one or two nodes (see node 2 in Fig. 1, for an example) that are not real communities and should be reassigned into other communities. Communities whose number of nodes below \( k_{\text{min}} \) are defined as small communities. Check the size of all communities, nodes belonging to small communities are reassigned to other communities which have most connections with.

**Step 6:** Identify communities by frequencies.
For all communities identified after Step 5, compute the frequencies of all nodes by (3) and obtain the frequency distribution curves (see Fig. 5(b), for an example). For simplicity, the ranges of the frequencies are identical for all communities. Let \( n_{\text{m}} \) denote the number of sub-intervals in the frequency distribution curve. For each curve, the nodes between two valleys are assigned to the same community, and the number of communities is the number of peaks in the distribution curve. In Fig. 5(b), two communities \( C_1 + C_2 \) are separated by the frequency distribution curve.

**Step 7:** Misclassified nodes detection in large communities.
Some nodes may be mistakenly classified into large communities by the previous steps, especially the method of reassignment of small communities in Step 5, which should be detected and reassigned. Let \( d_i(\text{in}) \) denote the number of in community links, then the ratio of in community links of node \( i \) is defined as

\[
\text{Ratio}_i = \frac{d_i(\text{in})}{N_i},
\]

where \( N_i \) is the number of nodes in the community. If \( \text{Ratio}_i \) is lower than the threshold \( \text{Ratio}_{th} \), node \( i \) is judged as a misclassified node. In the simulations of this paper, \( \text{Ratio}_{th} = 0.4 \).

**Step 8:** Small communities and misclassified nodes reassignment.
After Step 6, small communities may be generated again. Check the size of all communities; the nodes belonging to the small communities are reassigned to other communities which have most connections, along with the misclassified nodes detected in Step 7.

**D. Parameters selection**
There are four parameters in the algorithm: \( K_p \), \( K_n \), \( n_{\text{b}_{\text{p}}} \), and \( n_{\text{b}_{\text{n}}} \).

The value of \( K_p \) and \( K_n \) are applicable in a very wide range for community detection. The range varies along with the scale (\( N \)) of the network to be detected. For example, when \( N = 100 \) in the network, \( K_p \) and \( K_n \) can be assigned value in \([0.1, 10]\) and \([-0.1, -3.3]\), respectively; when the scale increased to \( N = 1000 \), \( K_p \) and \( K_n \) can be assigned value in \([0.1, 45]\) and \([-0.1, -3.3]\), respectively.

\( n_{\text{b}_{\text{p}}} \) and \( n_{\text{b}_{\text{n}}} \) also relate to the scale of the network. The smaller the two parameters are, the more communities are detected. They can be set around \( n_{\text{b}_{\text{p}}} = 50 \) and \( n_{\text{b}_{\text{n}}} = 500 \) when \( N = 100 \). For \( N = 1000 \), the two parameters can be set around \( n_{\text{b}_{\text{p}}} = 200 \) and \( n_{\text{b}_{\text{n}}} = 2000 \). The interval of the frequency is usually \([-0.2, 0.2]\).

\( k_{max} \) and \( k_{min} \) are the maximum and minimum degrees of nodes in the network. The DTD algorithm does not need true value of the degrees. An approximate value is enough, which will not exert an evident influence to the result.

**IV. EXPERIMENTS**
In this section, the performance of the proposed algorithm is estimated. Because the dynamics-based algorithm proposed in Ref. 37 is not applicable in networks with many (larger than three) communities. So three modularity based algorithms are selected to compare with the proposed algorithm. One is the Newman’s famous fast algorithm (FA),\(^{18}\) the second is the recent algorithm firstly implemented by Blondel, Guillaume, Lambiotte, and Lefebvre (BGLL) which is a variant of hierarchical agglomerative clustering approaches,\(^{46}\) and the third is the density shrinkage (DS) algorithm.\(^{47}\) The three algorithms are efficient and can run on large scale networks.

The LFR benchmark is used to generate various networks with different parameters. The scales of the networks are \( N = 5000 \) and \( N = 50000 \). For the DTD, \( K_p = 20 \) and \( K_n = -2 \) in all the simulations of this section. The values of \( n_{\text{b}_{\text{p}}} \) and \( n_{\text{b}_{\text{n}}} \) are related to the scale of the networks, \( n_{\text{b}_{\text{p}}} = 400 \) and \( n_{\text{b}_{\text{n}}} = 8000 \) when \( N = 5000 \), \( n_{\text{b}_{\text{p}}} = 2000 \) and \( n_{\text{b}_{\text{n}}} = 12000 \) when \( N = 50000 \). For each network scale, the networks have different mixing parameters ranging from 0.1 to 0.6.

The normalized mutual information (NMI) is used to evaluate the performance of the proposed algorithm.\(^{38}\) The value of NMI is limited in \([0, 1]\). If there is only one community detected, \( \text{NMI} = 0 \). If the detected communities are identical to the real communities, \( \text{NMI} = 1 \). The higher value of NMI indicates higher performance of the algorithm.

**A. Evaluation on networks of \( N = 5000 \)**
The experiments are designed similarly as that in Ref. 41, two groups of networks are tested corresponding to average degree \( \langle k \rangle = 15 \) and \( \langle k \rangle = 20 \), respectively. The maximum degree of the networks is \( k_{max} = 50 \). The values of the parameters \( \gamma \) and \( \beta \) are labeled in the caption of the figures. The simulation results are shown in Figs. 6 and 7, respectively. Each value of NMI is the average obtained on 30 networks generated with the same parameters. As shown in Figs. 6 and 7, the values of NMI obtained by the DTD are higher than the rest three algorithms in most cases. As the mixing parameter increases, the community structure
becomes indistinct. The difficulty of detecting community increases as the increasing of the mixing parameter $\mu$, so the values of $NMI$ decrease in all algorithms. In the cases of higher values of $\mu$, the advantage of the DTD is more evident. When the mixing parameter $\mu = 0.6$, the values of $NMI$ of the DTD still exceed 0.98 in all the cases shown in the figures. It is well known that it is more difficult to detect the communities when the average degree is lower (please see Figs. 5–8 in Ref. 41). The proposed algorithm is not as sensitive as the DS and BGLL to the decreasing of the average degree. In case that the average degree $\langle k \rangle = 15$ in Fig. 6, the proposed DTD obtains much better result than the rest three algorithms.

B. Evaluation on networks of $N = 50000$

Another important criterion to evaluate an algorithm for community detection is the efficiency, this can be tested on large scale networks, e.g., $N = 50000$. Fig. 8 is the simulation results of the FA, DS, BGLL, and DTD algorithms on networks with $N = 50000$. The average degree is $\langle k \rangle = 30$ and the maximum degree is $k_{max} = 100$. The values of the parameters $\gamma$ and $\beta$ are labeled in the caption of Fig. 8. The values of $NMI$ obtained by the DTD always maintain near 1, which is better than the FA, DS, and BGLL. Similar as that on networks of $N = 5000$, the larger the value of $\mu$, the more evident the advantage of the DTD is. When $\mu$ exceeds 0.4, the $NMI$ of the FA falls below 0.6 and BGLL falls below 0.9, while the proposed DTD remains 0.99.

C. Run time evaluation

In this subsection, the run time of the four algorithms is tested on the LFR benchmark networks in different network sizes from $N = 100$ to $N = 50000$. Fig. 9 shows the simulation results. Each point in Fig. 9 is the average on 10
V. CONCLUSION

A discrete-time clustering model for modular networks is proposed, which acts as a bridge between the structure and the dynamics of the network. For a modular network, phase clustering phenomenon is observed from the model; on the other hand, from the phase clustering phenomenon, the communities of the network can be detected. Based on the proposed model, the DTD algorithm for community detection is also proposed. The performance and efficiency of DTD are verified by comparative experiments on large scale networks.

Only the stable phase information is applied in community detection in the previous dynamics based algorithms, including synchronization and clustering. Based on the proposed model, both the phase information and theirs evolving frequencies are applied for community detection, thus the proposed DTD algorithm is of high performance. We use difference equation instead of differential equation, thus DTD is efficient and can run on large scale networks.

networks. The time complexity of the Newman’s FA is $O(mN)$, where $m$ is the number of edges. The time complexity of the DS is $O(m\log(N))$.\(^47\) The most computational time of the DTD is costed in the iteration of (2) with complexity $O(N^2)$, which coincides with the simulation results shown in Fig. 9.

![Fig. 8. Simulation results on networks with $N = 50,000$. (a) $(\gamma, \beta) = (2, 1)$. (b) $(\gamma, \beta) = (2, 2)$. (c) $(\gamma, \beta) = (3, 1)$. (d) $(\gamma, \beta) = (3, 2)$.](image1)

![Fig. 9. Run time of the four algorithms with different sizes of the LFR networks from $N = 100$ to $N = 50,000$, each point is the average on 10 networks with the same parameters.](image2)

34S. Liu and M. Zhan, Chaos 24, 013104 (2014).