A thermal flux-diffusing model for complex networks and its applications in community structure detection*

Shen Yi(沈 毅)†

College of Information Science and Technology, Nanjing Agricultural University, Nanjing 210095, *China*

(Received 4 June 2012; revised manuscript received 19 October 2012)

We introduce a thermal flux-diffusing model for complex networks. Based on this model, we propose a physical method to detect the communities in the complex networks. The method allows us to obtain the temperature distribution of nodes in time that scales linearly with the network size. Then, the local community enclosing a given node can be easily detected for the reason that the dense connections in the local communities lead to the temperatures of nodes in the same community being close to each other. The community structure of a network can be recursively detected by randomly choosing the nodes outside the detected local communities. In the experiments, we apply our method to a set of benchmarking networks with known pre-determined community structures. The experiment results show that our method has higher accuracy and precision than most existing globe methods and is better than the other existing local methods in the selection of the initial node. Finally, several real-world networks are investigated.

Keywords: complex networks, community structure, thermal flux-diffusing model

PACS: 89.75.Hc, 89.75.Da **DOI:** 10.1088/1674-1056/22/5/058903

1. Introduction

The complex network is an effective tool for studying real systems such as social networks,[1] metabolic networks,[2] and citation networks.[3] Since the topological information on such complex networks is becoming available through computer-aided techniques, there is growing interest in the investigation of the properties of such networks.[4–8] It is observed that the complex networks are structured in terms of modules and communities, which are groups of nodes defined based on different concepts.[9–11] Since the communities in networks is closely related to the functional units in various fields ranging from the social groups in social networks[1] to the sets of pages on related topics on the world-wide web,[12] the prosperities of the community structure are indeed of considerable importance and have attracted great interests.

Before detecting communities in a network, the first thing is to look for a quantitative definition of community. Because the definition of community often depends on the specific system at hand or the application in mind, no universal definition is found.[11] Different definitions such as basics local definitions, globe definitions, and others based on the node similarity are proposed and discussed.[11] In fact, for binary networks, an acceptable definition of community is the groups of nodes whose edges within the same group are much denser than those between different groups. This community structure reflects well on the close connections between the units. In order to detect the communities in binary networks, many methods, which usually are divided into the traditional methods, divisive algorithms, modularity Q-based methods,[9,10] dynamic algorithms, and so on,[11] have been proposed. Most of those methods aim at detecting the standard partitions, i.e., the partitions in which each node is assigned to a single community. However, in real networks, nodes are often shared among communities, which leads to overlapping communities.[13] In the last few years, the overlapping-community detecting has become quite popular.[13,14] Besides overlapping characteristic, the community structure of real world networks often exhibits multiple scales.[15,16] In this case, a network can be divided at different hierarchical levels, which constitute different topological descriptions. Recently, a general framework has been proposed for the detection of community structure in multi-scale networks.[15] Another significant issue not mentioned in the above methods is how to directly detect the local community that encloses a given node, which is meaningful when all the communities of a graph are not completely known. Recently, several efficient local methods such as the random walk-based method[17] and the Bagrow method[18] have been proposed for detecting the local community enclosing a given node. However, the computation is usually highly complex,[18] or the accuracy is sensitive to the pre-defined parameters such as the iteration walking steps.[17]

Based on the thermal flux diffusing mechanism, we introduce a physical method to find the local community that encloses a given node without the information of the entire community structure in this paper. In our method, the temperature of nodes increases, which is consistent with the principle of flux diffusing, and the dense connections lead to the tem-

*Project supported by the National Natural Science Foundation of China (Grant No. 60672095), the Fundamental Research Funds for the Central Universities, China (Grant No. KYZ201300), and the Youth Sci-Tech Innovation Fund of Nanjing Agricultural University, China (Grant No. KJ2010024).

†Corresponding author. E-mail: shen_yi1979@njau.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd

perature of nodes in the same community being close to each other. Therefore, the local community enclosing a given node that has a high temperature can be detected accurately. For a network with n nodes and m edges, the community enclosing a given node can be detected in time $O(mt/\Delta t)$ if the diffusion time and the integral interval are set to be t and Δt, respectively. Based on this mechanism, the community structure of a network can be detected in a recursively. For a network with c communities, the community structure can be detected in time $O((cmt/\Delta t)+n)$. In the experiments, we first test the accuracy and the precision of our method on 128-node ad hoc networks. and then apply our method to the BA model, a benchmarking network, which is designed specifically for testing local methods and processes a wider degree distribution. Several existing local methods such as the random walk-based method,[17] the Clauset method,[19] and the Bagrow method[18] are also investigated. Finally, we apply our method to several real-world networks.

2. The thermal flux model and the local community detecting method

In this section, we first introduce the thermal flux model of networks, and then present our physical method based on this model for detecting communities.

2.1. Thermal flux model of networks

We first describe our model by using a simple network. Without loss of generality, we focus on a simple case in which two nodes have only one edge between them if they are connected. The model is shown in Fig. 1, from which we can see that node i is connected with node j by edge l_k. We define edge l_k as an element with density ρ. At time t, the temperature at location s on edge l_k is T_s and is denoted by the function $T_s = T_s(s,t)$. The thermal flux at location s is q_s, and the thermal flux from the outside of the network is $\tilde{q}_s$. The velocity of the thermal flux is V_s.

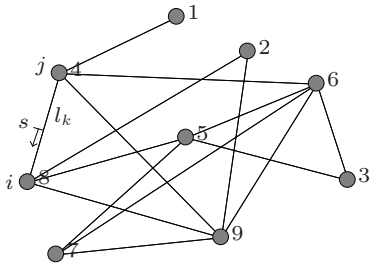


Fig. 1. Model based on thermal flux diffusing.

According to the theory of thermodynamics that the thermal energy is conserved,[20] the thermal dynamic equation of the point at location s on edge l_k can be written as

$$\rho c\left(\frac{\partial T_s}{\partial t}+V_s\frac{\partial T_s}{\partial l_k}\right)+\frac{\partial T_s}{\partial l_k}q_s=\tilde{q}_s, \tag{1}$$

where c is the heat coefficient. Then, Fourier's law is used to link heat flux q_s and the thermal gradients, In this way, q_s can be given as

$$q_s=-K\frac{\partial T_s}{\partial l_k}, \tag{2}$$

where K is the conductivity coefficient of edges, ρ, c, V_s, and K are assumed to be constants and are set to be 1 here for simplicity. Combined with Eq. (2), Eq. (1) can be written as

$$\frac{\partial T_s}{\partial t}+\frac{\partial T_s}{\partial l_k}-\frac{\partial}{\partial l_k}\left(\frac{\partial T_s}{\partial l_k}\right)=\tilde{q}_s. \tag{3}$$

For a network with m edges, pre-multiplying Eq. (3) with a virtual change in temperature δT_s, we can write the dynamic equation of the network as

$$\int_m\left(\delta T_s\left(\frac{\partial T_s}{\partial t}+\frac{\partial T_s}{\partial l_k}\right)-\frac{\partial}{\partial l_k}\left[(\delta T_s)\left(\frac{\partial T_s}{\partial l_k}\right)\right]\right)\mathrm{d}\,(m)$$
$$=\int_m(\delta T_s)\tilde{q}_s\,\mathrm{d}\,(m), \tag{4}$$

then the temperature at any location s on edge l_k in Fig. 1 is

$$T_s=\frac{1}{2}\left\{\begin{array}{c}(1-s)\\(1+s))\end{array}\right\}^{\mathrm{T}}\left\{\begin{array}{c}T_i\\T_j\end{array}\right\}=\boldsymbol{S}_k^{\mathrm{T}}\boldsymbol{T}_k, \tag{5}$$

where $\boldsymbol{S}_k$ is the temperature distribution functions of edge l_k. And $T_s=T_i$ when $s=-1$, $T_s=T_j$ when $s=1$. The time derivative of Eq. (5) can be rewritten as

$$\dot{T}_s=\frac{\partial T_s}{\partial t}=\boldsymbol{S}_k^{\mathrm{T}}\dot{\boldsymbol{T}}_k, \tag{6}$$

and δT_s has the form $\delta T_s=\boldsymbol{S}_k^{\mathrm{T}}\delta\boldsymbol{T}_k$. Letting $\boldsymbol{B}=\partial(\boldsymbol{S}_kT_s)/\partial l_k$ and combining the variational statement of Eq. (4) with Eqs. (5) and (6), we have

$$\int_m\left((\delta\boldsymbol{T}_k)^{\mathrm{T}}\boldsymbol{S}_k\boldsymbol{S}_k^{\mathrm{T}}\dot{\boldsymbol{T}}_k+(\delta\boldsymbol{T}_k)^{\mathrm{T}}\boldsymbol{S}_k\boldsymbol{B}\boldsymbol{T}_k\right)\mathrm{d}\,(m)$$
$$-\int_m(\delta\boldsymbol{T}_k)^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{T}_k\,\mathrm{d}\,(m)=\int_k(\delta\boldsymbol{T}_k)^{\mathrm{T}}\boldsymbol{S}_k\tilde{q}\,\mathrm{d}(m), \tag{7}$$

where $\boldsymbol{T}_k$, $\dot{\boldsymbol{T}}_k$, and $\delta\boldsymbol{T}_k$ are nodal quantities that do not change over the edge and can be removed from the integral. After throwing away the pre-multiplied arbitrary vector $\delta\boldsymbol{T}_k$, we can reduce Eq. (7) to a dynamical equation as

$$\boldsymbol{C}_k^{\mathrm{t}}\dot{\boldsymbol{T}}_k(t)+\boldsymbol{K}_k^{\mathrm{t}}\boldsymbol{T}_k(t)=\boldsymbol{Q}_k(t), \tag{8}$$

where

$$\boldsymbol{C}_k^{\mathrm{t}}(t)=\int_m\boldsymbol{S}_k\boldsymbol{S}_k^{\mathrm{T}}\,\mathrm{d}\,(m), \tag{9}$$

$$\boldsymbol{K}_k^{\mathrm{t}}(t)=\int_m(\boldsymbol{S}\boldsymbol{B}-\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})\,\mathrm{d}\,(m), \tag{10}$$

$$\boldsymbol{Q}_k(t)=\int_m\boldsymbol{S}_k\tilde{q}\,\mathrm{d}\,(m)=\int_m-\boldsymbol{S}_k\boldsymbol{B}\boldsymbol{T}_{ck}(t)\,\mathrm{d}\,(m), \tag{11}$$

here $\boldsymbol{T}_{ck}(t)=\left[\begin{array}{cc}T_{ci}(t) & T_{cj}(t)\end{array}\right]$, $T_{ci}(t)$ is the temperature load for node i from the outside of the network at time t (except for the

source node which has a high temperature, $T_{ci}(t) = 0$). The temperature of nodes at any time t can be obtained by reading the temperature vector $\boldsymbol{T}_k$ of the nodes. For a network with n nodes and m edges, the computational complexity of generating the model is $O(m)$, and the number of iterations is $O(t/\Delta t)$ if the diffusion time and the integral interval are set to be t and Δt, respectively. Hence, the temperature of all nodes can be obtained in time $O(mt/\Delta t)$.

2.2. Method

From the thermal flux model, we can see that if we provide a high temperature load for a selected node and low temperature loads for the other nodes, the thermal flux will diffuse among the network naturally, and the temperatures of nodes will change. Because there are dense connections among the local communities, the temperatures of the nodes in the same community will be close to each other. Based on this mechanism, the local community enclosing a given node with the high temperature can be detected accurately. We present the detailed implementation of our method as follows.

(i) We construct the thermal flux model of the network.

(ii) Provide a high temperature load for a given node and low temperature loads for the other nodes.

(iii) Set the diffusing time as t and the integral interval as Δt, and calculate the temperatures of all nodes.

(iv) Agglomerate the node with the temperature that is most close to the average temperature of the divided group of nodes (which is composed of the given node at first) until the local maximal modularity[19] is obtained. Then the local community enclosing the given node is detected.

(v) Select a new node outside the detected local communities, and then go to step (ii).

Our method is a thermal flux-based method and the temperature of each node is calculated based on the greed principal of thermal flux diffusing. The maximal increment of the local modularity can be obtained for the reason that our method agglomerates the node whose temperature is closest to the average temperature of the divided group of nodes each time, which makes our method have a high accuracy. Our method works recursively until all local communities are detected. If a network has m edges and c communities, and the diffusing time and the integral interval are set to be t and Δt, respectively, the community structure can be detected in time $O((cmt/\Delta t)+n)$ as our method only recursively runs c times and agglomerates n nodes totally. The computational complexity of the method scales linearly with the network size and is lower than most existing techniques.[5,11]

3. Experiment

In this section, we first compare our method with several global methods on detecting the community structure of a network, and then compare our method with several local methods on detecting the local communities. Finally, we apply our method to several real-world networks. For our method, the temperature of the given node, the temperature of the other nodes, the diffusing time t, and the integral interval Δt are set to be 1 , 0, 1.5, and 0.25, respectively.

3.1. Comparison with global methods

3.1.1. Analysis of accuracy

We generate a set of 128-node ad hoc networks, which have been widely accepted as the standard benchmarking networks for investigating the accuracy of community structure detection methods,[5,11] to investigate the accuracy of our method. Each network has 128 nodes consisting of four communities of 32 nodes each. Edges are placed independently and randomly between the node pairs. Each node has z_{in} links on average to its fellows in the same community, and z_{out} links to the outside, and $z_{\text{in}} + z_{\text{out}} = 16$. In Fig. 2, we show the fraction of nodes classified correctly into four communities. We also make comparisons with two recent methods (the extremal optimization (EO) algorithm[21] and the simulated annealing (SA) algorithm[22]) in terms of accuracy.

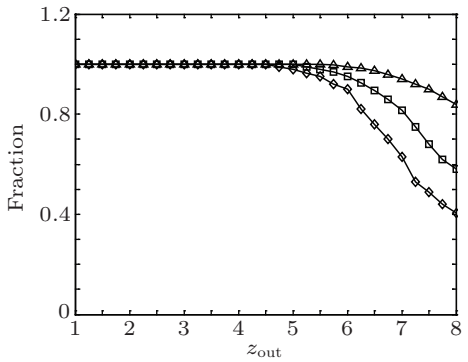


Fig. 2. Fractions of nodes correctly identified by our method (squares), the EO algorithm (diamonds), and SA algorithm (triangles) averaged over 200 networks as a function of z_{out}.

From Fig. 2 we can see that all methods perform very well from $z_{\text{out}} = 0$ to around $z_{\text{out}} = 6$ and can identify more than 90% of all nodes correctly. When $z_{\text{out}} \geq 6$, the classification begins to deteriorate markedly. The random network has been used by many researchers to test their algorithms,[5,11,23,24] among which the SA algorithm performs the best. For $z_{\text{out}} = 8$, it can identify more than 80%[5,22] of the nodes correctly. However, the SA is a parameter-dependent and high computation complexity algorithm,[5,11] so it is not a reasonable method for large networks. Our method can identify more than half the nodes correctly when $z_{\text{out}} = 8$, and is higher than most others (the details can be seen in a recent comparison in Ref. [5]).

3.2. Analysis of precision

We define the precision as the consistence among the community structures obtained from different runs of an al-

gorithm on the same network.[25] The normalized mutual information measurement (NMI)[25] proposed by Danon *et al.* is used to quantitatively measure the precision of an algorithm. The NMI is based on a confusion matrix $\boldsymbol{N}$, where the element N_{ij} is the number of nodes in the "real" community i that appear in the "found" community j. Here the "real" and "found" communities are the communities detected by two runs of an algorithm on the same network respectively. The NMI(A,B) is the measure of the similarity between partitions A and B. It is obvious that it needs more runs of an algorithm on the same network for effectively measuring the precision of an algorithm. We propose a matrix $\boldsymbol{M}$ where the element $M_{ij} = M(i,j)$ accounts for the similarity between the i-th and the j-th detected community structures. If we run k times of an algorithm, the precision denoted by P is written as

$$P = \frac{\sum_{i}^{k} \sum_{j(j\neq i)}^{k} M_{ij}}{k^2 - k}. \tag{12}$$

The denominator accounts for the summation of M_{ij}, and the numerator accounts for the total number of M_{ij} in the denominator. If the community structures detected by different runs of an algorithm are identical to each other, P will take its maximum value 1. If they are independent of each other, P will take its minimal value 0.

We now use the P function to investigate the precision of our method. We generate 200 networks according to the approach described in Subsection 3.1, and then run each algorithm 100 times to find the communities in each network. The value of P is obtained by averaging over 200 networks and 100 runs on each network. The results of the precision are shown in Fig. 3. We also make comparison with two recent methods, the EO algorithm[21] and the SA algorithm.[22]

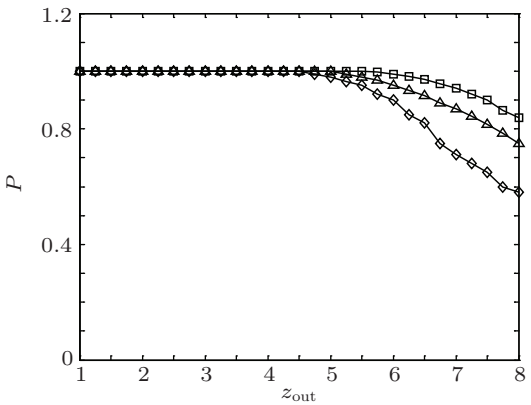


Fig. 3. Precision of our method (squares), the EO algorithm (diamonds), and the SA algorithm (triangles) as a function of z_{out}. Each point is obtained by averaging over 200 networks and 100 runs on each network.

From Fig. 3 we can see that the differences among the results of the three methods are small for small z_{out}. However, the precision of our method is obviously higher than that of the other two methods when $z_{out} > 6$. The reason is that the EO algorithm uses a heuristic search to optimize the modularity Q[9] by dividing the network into groups, however the probability of moving the same edges decreases with the increase of z_{out} in different runs of the EO algorithm on the same network. Different from the EO algorithm, our method is a thermal-flux-based method and the network topology completely determines the correlation between the temperatures of the nodes, which means that our method can divide the same nodes into the same groups more easily than the EO algorithm. Therefore, for large z_{out}, our method performs better than the EO algorithm in terms of the precision. The SA method is based on finding the ground-state energy of a Hamiltonian.[22] Because this Hamiltonian corresponds to an S-state Potts model with both ferromagnetic and anti-ferromagnetic terms, and two-, three-, and four-spin interactions like the graph coloring problem,[22] the precision can not be ensured when the network is fuzzy with large z_{out}.

3.3. Comparisons with local methods

Recently, several typical local methods have been proposed, such as the random walk-based method,[17] the Clauset method,[19] and the Bagrow method.[18] The idea behind the random walk-based method is that the probability for an agent starting from any node and staying in the same community after a number of transitions is larger than that of going out to a different community. Hence, the accuracy of this method is sensitive to the iteration walking steps.[17] The Clauset and the Bagrow methods begin with an initial node, and one or more nodes are chosen and agglomerated at each step until appropriate stopping criteria[18,20] have been satisfied. Then, the local community enclosing the starting node is obtained. However, the two methods still have high computation complexities. The Clauset method runs in time $\mathrm{O}(|C|^2 d)$,[19] and the Bagrow method runs in time $\mathrm{O}(|C|d^2 \log|B|)$,[18] where d is the average degree of the network, C is the number of nodes in the local community, and B is the set of nodes adjacent to C which have at least one neighbor in C.

The accuracy of the found communities is traditionally quantified by the fraction of correctly identified nodes. However, it has drawbacks,[26] and the binary partitioning of a local algorithm has further problems.[18] Therefore, we use the normalized mutual information[27] to measure how well the true partition P_{R} and the found partition P_{F} match to each other, which is written as[18,27]

$$I(P_{\mathrm{R}}, P_{\mathrm{F}}) = \frac{-2\sum_i \sum_j X_{ij} \log\left(\frac{X_{ij}N}{X_{i.}X_{.j}}\right)}{\sum_i X_{i.} \log\left(\frac{X_{i.}}{N}\right) + \sum_j X_{.j} \log\left(\frac{X_{.j}}{N}\right)}, \tag{13}$$

where $\boldsymbol{X}$ is a 2×2 matrix with X_{ij} being the number of nodes from real group i that are placed in found group j,

$X_{\cdot j} = X_{1j} + X_{2j}$, and $X_{i\cdot} = X_{i1} + X_{i2}$. The $I(P_R, P_F)$ measures how much is known about partition P_R by knowing partition P_F. When $I = 1$, P_R and P_F are identical to each other, while they completely different from each other when $I = 0$.

The 128-node ad hoc networks have a sharply peaked degree distribution, and each node has the same degree and status in the network. For local methods that detect the community enclosing a given node, we should use more realistic networks which should possess a wider degree distribution, such as a power-law distribution. To do this, we use the BA model.[28] Based on the BA model, the benchmarking network is constructed as follows.[18] We first use a BA model of $N = 512$ and $m_0 = m = 8$ to generate a graph G, and then randomly partition the nodes of G into four equally sized groups, which will be served as the actual communities. Finally, we choose random pairs of edges that are between the same two groups, and rewire them to be within the groups in such a way that the degree distribution is unaltered. With the increase of the number of rewriting r, the communities are well separated. Different from the 128-node ad hoc networks, the BA model processes a wider degree distribution. The degree of the starting node inevitably has an influence on the accuracy. Therefore, we test our method on the BA model by choosing leaf nodes, hub nodes, and nodes at random as the starting nodes respectively.

In order to terminate a local method and find the local community, we should use stopping criteria. In our tests, we use the stopping criteria called the best of $\{p\}$-strong criteria, which were proposed by Bagrow.[18] The criteria are proposed based on the concept of a strong community,[29] in which every node has more neighbors inside than outside

$$k_i^{\text{in}}(C) > k_i^{\text{out}}(C), \quad \forall i \in C. \tag{14}$$

It is obvious that we should stop agglomerating nodes if the community is no longer strong. However, this requirement is too strong. Hence, a weaker requirement will be employed. A community is p-strong if equation (14) holds only for a fraction p of the nodes in C

$$\sum_{i \in C} \left[k_i^{\text{in}}(C) > k_i^{\text{out}}(C) \right] \geq p|C|. \tag{15}$$

A set of appropriate parameters, $\{p_1, p_2, \ldots\}$, can be used simultaneously for a local algorithm. When the stopping criteria of each p_i are satisfied, we should maintain the state of C. Next, we can use the local modularity R[20] to choose the best corresponding $C(p_i)$, i.e., finding the C with the largest R. It is also obvious that a small p (e.g. $p = 0.1$) can lead to the stopping very quickly. In our tests, we use $\{p\} = \{0.75, 0.76, \ldots, 1.0\}$, the interval is small enough to produce a correct local community. The stopping criteria are applicable for the agglomerative methods, such as the Clauset method, the Bagrow method, and our method. The random walk-based method stops when the difference between two consecutive sorted node sequences converges as the iteration walking step l[17] increases. Figure 4 shows the results of these methods on the BA model.

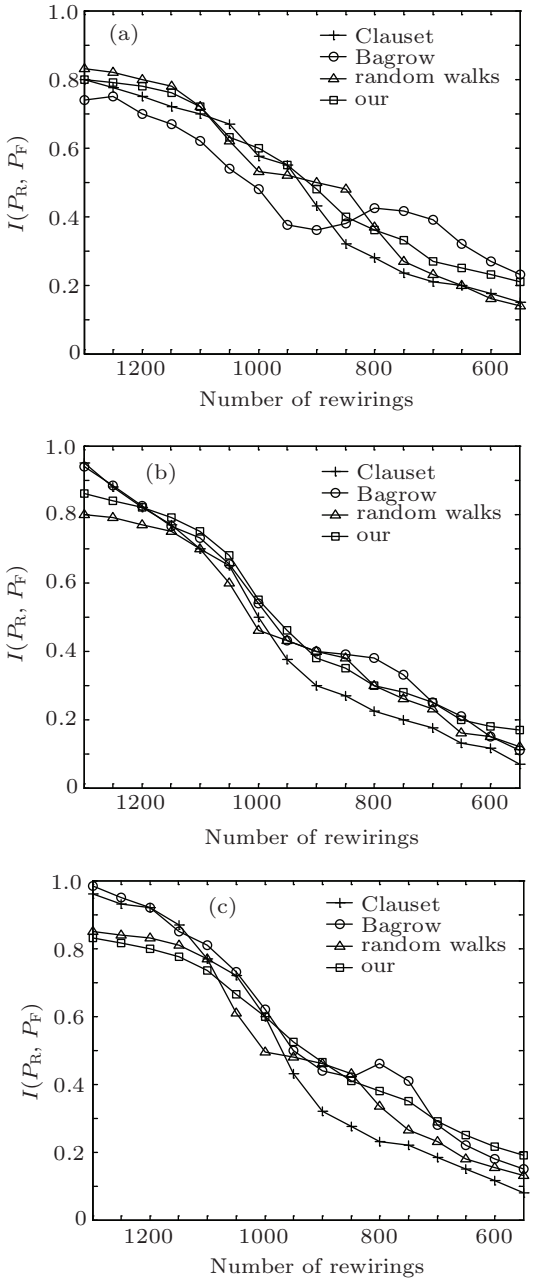


Fig. 4. Tests on the BA model, averaged over 600 realizations. Results are obtained by the Clauset method, the Bagrow method, the random walk-based method, and our method as a function of the number of rewritings r. Initial selected nodes are hub nodes, leaf nodes, and randomly selected nodes in panels (a), (b), and (c), respectively.

From Fig. 4 we can see that the accuracies of all methods decrease with the increase of the number of rewritings, the reason is that the community structure with a small number of rewritings is fuzzy. In the case that the initial node is a hub node, the random walk-based method and our method perform better than the other two methods when the number of rewritings is larger than 1100 (in other words, the network

is clearly separated). With the decrease of number of rewritings, our method performs better than the random walk-based method. In the case that the initial node is a leaf node, the accuracy of our method is slightly higher than that of the random walk-based method and the Clauset method for a small number of rewritings. In the case that the initial node is a randomly-selected node, the Bagrow method performs best in most cases except for the very small number of rewritings case. In conclusion, we observe that the performance of our method is better than that of the other three methods for the selection of the initial node. Furthermore, we observe that a randomly selected initial node usually corresponds to a higher accuracy when the number of rewritings is large (communities are clearly separated). This means that the nodes with medium degrees are good starting nodes for a local method when the network is clearly separated. However, with a small number of rewritings (as the separation blurs), starting from a hub node allows for obtaining more accurate results.

3.4. Test on real networks

3.4.1. The Zachary network

Now we apply our method to the well-known karate club network,[30] which was studied by Zachary and has become a commonly workbench used for evaluating the community-detecting algorithm.[11] The network consists of 34 members of a karate club and 78 edges representing the friendship among the members of the club. Many existing methods divide the club into two communities due to a disagreement between the administrator and the instructor of the club.[11] The results of our method are shown in Fig. 5.

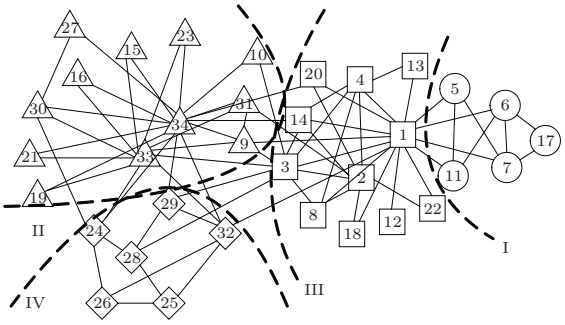


Fig. 5. Result of the karate club network obtained by our method. Different shapes of nodes represent different communities in the network. The detecting sequence of local communities is also plotted.

According to the sequence of random-selected nodes, our method can detect four communities. Our result is different from the division reported by Zachary. Considering the groups obtained after each division as local communities and calculating the modularity Q, we can examine the accuracy of the recursive detecting process of our method. The Q increases to its maximal value $Q = 0.44$, which is larger than that of most existing methods[5] and is the same as that of the SA algorithm.[22] It means that our method has a high accuracy since the larger modularity corresponds to the higher accuracy.

3.4.2. Collaboration network

As another real example, we apply our method to a collaboration network,[31] which contains 126 nodes representing the research members. Two nodes are connected if they coauthored articles during the year 1995. We investigate the partition of this network into groups that represent communities by using our method. The maximal Q is 0.562 achieved for finding 15 communities corresponding to 15 research groups, which is the same as the result of other methods.[1,9,21] The local communities (denoted by the node number) are detected in the sequence of (43–49) → (1–6) → (58–61) → (36–42) → (83–88) → (7–13) → (89–96) → (62–70) → (50–57) → (24–35) → (109–117) →(14–23) → (97–107) → (118–126) → (71–82), and the community structure of the collaboration network is shown in Fig. 6.

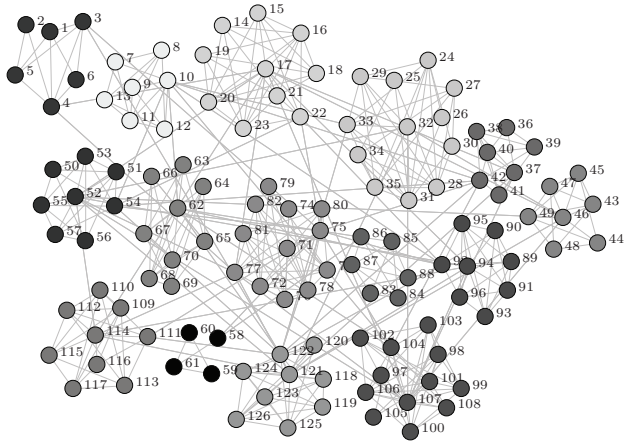


Fig. 6. Community structure of a collaboration network detected by our method.

3.4.3. Other real networks

Finally, as a further test of our algorithm, we apply our method to two large real-world networks. The first is the US college football network,[1] 115 nodes represent teams (identified by their college names), and 613 edges represent regular-season games between two teams. This network is interesting since it incorporates a known community structure, which makes it a commonly used workbench for evaluating the community-detecting algorithm.[5] Ten communities which contain ten conferences are identified by our algorithm. The group is classified the same as the results in Refs. [9] and [32]. The second is a large friendship network. The network is constructed from the in-school questionnaires of the Add Health[31] friendship nomination study during the years 1994 and 1995, in which 75871 students in the US are investigated. We detect five main communities: white, black, Hispanic, Asian, and mixed, which are approximately the same as the race classification.

4. Conclusion

In this paper, we introduce a thermal flux-diffusing model for complex networks, based on which we propose a physical method to detect the local communities in the complex networks. For a randomly given node, our method can find the local communities that enclose the randomly given node without the knowledge of the entire network. For a network with n nodes and m edges, the local community enclosing a given node can be detected in time $\mathrm{O}(mt/\Delta t)$ if the diffusion time and the integral interval are set to be t and Δt, respectively. All communities can be detected in a recursive way. Therefore, the community structure of the network can be detected in time $\mathrm{O}((cmt/\Delta t)+n)$ if it has c communities, as our method only recursively runs c times and agglomerates n nodes totally. The computational complexity of our method scales linearly with the network size and is lower than most existing techniques.[5,12] In the experiments, we apply our method to a set of benchmarking networks with known pre-determined community structures. The experiment results show that our method has higher accuracy and precision than most existing globe methods and is more robust than other existing local methods to the selection of the initial node. Finally, we apply our method to several real-world networks as a further test of our algorithm. The communities detected by our method clearly reveal the functional groups in the networks.

References

[1] Girvan M and Newman M E J 2002 *Proc. Natl. Acad. Sci. USA* **99** 7821
[2] Palla G, Derényi I, Farkas I and Vicsek T 2005 *Nature* **435** 814
[3] Redner S 1998 *Eur. Phys. J. B* **4** 131
[4] Chen D B, Shang M S, Lv Z H and Fu Y 2010 *Physica A* **389** 4177
[5] Lancichinetti A and Fortunato S 2009 *Phys. Rev. E* **80** 056117
[6] Wang X H, Jiao L C and Wu J S 2010 *Chin. Phys. B* **19** 020501
[7] Zhang B D, Tang Y H, Wu J J and Li X 2011 *Chin. Phys. B* **20** 098901
[8] Fan C X, Wan Y H and Jiang G P 2012 *Chin. Phys. B* **21** 020510
[9] Newman M E J and Girvan M 2004 *Phys. Rev. E* **69** 026113
[10] Newman M E J 2006 *Phys. Rev. E* **74** 036104
[11] Fortunato S 2010 *Physics Reports* **486** 75
[12] Flake G W, Lawrence S R, Giles C L and Coetzee F M 2002 *IEEE Computer* **35** 66
[13] Shen H W, Cheng X Q, Cai K and Hu M B 2009 *Physica A* **388** 1706
[14] Ahn Y Y, Bagrow J P and Lehmann S 2010 *Nature* **466** 761
[15] Mucha P J, Richardson T, Macon K, Porter M A and Onnela J P 2010 *Science* **328** 876
[16] Shen H W, Cheng X Q and Fang B X 2010 *Phys. Rev. E* **82** 016114
[17] Yang B, Cheung W K and Liu J M 2007 *IEEE T. Knowl. Data En.* **19** 1333
[18] Bagrow J P 2008 *J. Stat. Mech.* P05001
[19] Clauset A 2005 *Phys. Rev. E* **72** 026132
[20] Cengel Y A 2007 *Introduction to Thermodynamics and Heat Transfer + EES Software* (New York: McGraw Hill Higher Education Press) pp. 253–262
[21] Duch J and Arenas A 2005 *Phys. Rev. E* **72** 027104
[22] Guimerà R, Pardo M S and Amaral L A N 2004 *Phys. Rev. E* **70** 025101
[23] Zou S R, Peng Y J, Liu A F, Xu X L and He D R 2011 *Chin. Phys. B* **20** 018902
[24] Shen Y 2011 *Chin. Phys. B* **20** 040511
[25] Danon L, Guilera A D, Duch J and Arenas A 2005 *J. Stat. Mech.* P09008
[26] Danon L, Guilera A D, Duch J and Arenas A 2006 *J. Stat. Mech.* P11010
[27] Strehl A and Ghosh J 2002 *Journal of Machine Learning Research* **3** 583
[28] Albert R and Barabási A L 1999 *Science* **286** 509
[29] Radicchi F, Castellano C, Cecconi F, Loreto V and Parisi D 2004 *Proc. Natl. Acad. Sci. USA* **101** 2658
[30] Zachary W W 1977 *J. Anthropol. Res.* **33** 452
[31] Frank K A 1996 *Soc. Networks* **18** 93
[32] Reichardt J and Bornholdt S 2004 *Phys. Rev. Lett.* **93** 218701