

## Research Article

# Local Community Detection Algorithm Based on Minimal Cluster

**Yong Zhou, Guibin Sun, Yan Xing, Ranran Zhou, and Zhixiao Wang**

*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008, China*

Correspondence should be addressed to Yan Xing; [xingyan\\_cumt@163.com](mailto:xingyan_cumt@163.com)

Received 14 June 2016; Accepted 11 October 2016

Academic Editor: Wu Deng

Copyright © 2016 Yong Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to discover the structure of local community more effectively, this paper puts forward a new local community detection algorithm based on minimal cluster. Most of the local community detection algorithms begin from one node. The agglomeration ability of a single node must be less than multiple nodes, so the beginning of the community extension of the algorithm in this paper is no longer from the initial node only but from a node cluster containing this initial node and nodes in the cluster are relatively densely connected with each other. The algorithm mainly includes two phases. First it detects the minimal cluster and then finds the local community extended from the minimal cluster. Experimental results show that the quality of the local community detected by our algorithm is much better than other algorithms no matter in real networks or in simulated networks.

## 1. Introduction

Community detection on complex networks has been a hot research field. Recently, a large number of algorithms for studying the global structure of the network are proposed, such as the modularity optimization algorithms [1, 2], the spectral clustering algorithms [3–6], the hierarchical clustering algorithms [7–10], and the label propagation algorithms [11–14]. However, with the continuous expansion of complex networks, it is easy to collect large network dataset with millions of nodes. How to store such a large-scale dataset in computer memory to analyze is a huge challenge for scholars. The calculation for studying the overall structure of this kind of large-scale networks is unimaginable. So local community detection becomes an appealing problem and has drawn more and more attention [15–18]. The main task of local community detection is to find a community using the local information of the network. Local community detection has good extensibility. If the local community detection algorithm is iteratively executed, more local communities can be found and the whole community structure of the network can be obtained. The time complexity of this kind of global community detection algorithm is dependent on the efficiency and accuracy of local community detection algorithms, so the research of local community detection

algorithm still has a long way to go. There are several problems that need to be solved in the research of local community detection. First, we should determine the initial state and find the initial node for local community detection, so as to determine the needed local information; then, we need to select an objective function, and through continuous iterative optimization of the objective function we find the community structure with high quality; after that we need to find a suitable node expansion method, so that the algorithm can extract the local community from the initial state step by step; finally, in order to terminate the algorithm, a suitable termination condition is needed to determine the boundary of the community.

Most of local community detection algorithms are based on the above-mentioned process. The definition of local community detection is to find the local community structure from one or more nodes, but most of the existing local community detection algorithms, including Clauset [15], LWP [16], and LS [17], are starting from only one initial node. They greedily select the optimal nodes from the candidate nodes and add them into the local community. LMD [18] algorithm extends not from the initial node but from its closest and next closest local degree central nodes. It discovers a local community from each of these nodes, respectively. It still starts from single node and discovers

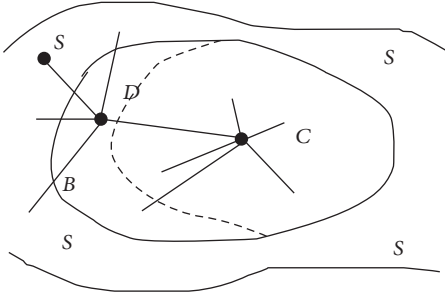


FIGURE 1: Definition of local community.

many local communities for the initial node. In general, the aggregation ability of a single node is lower than that of multiple nodes. So we do not just rely on the initial node as the beginning of local community expansion. Our primary goal is to find a minimal cluster closely connected to the initial node and then detect local community based on the minimal cluster. This can avoid instability because of the excessive dependence on the initial node. In this paper, we introduce a local community detection algorithm based on the minimal cluster—NewLCD. In this new algorithm, the beginning of community expansion is no longer from the initial node only, but a cluster of nodes relatively closely connected to the initial node. The algorithm mainly consists of two parts: one is the detection of the minimal cluster, and the other is the detection of the local community based on the minimal cluster. At the same time, the algorithm can be applied to the global community detection. After finding one local community using this algorithm, we can repeat the process to obtain the global community structure of the whole network.

## 2. Related Works of Local Community Detection

**2.1. Definition of Local Community.** The problem of local community detection is proposed by Clauset [15]. Usually we define the local community problem in the following way: there is a nondirected graph  $G = (V, E)$ ,  $V$  represents the set of nodes, and  $E$  represents the edges in the graph. The connecting information of partial nodes in the graph is known or can be obtained. The local community is defined as  $D$ . The set of nodes connected with  $D$  is defined as  $S$  and the set of nodes in  $D$  connected with nodes in  $S$  is defined as the boundary node set  $B$ . That is to say, any node in  $B$  is connected to one node in  $S$ , and the rest of  $D$  is the core node set  $C$ , as shown in Figure 1.

Local community detection problem is to start from a preselected source node. It adds the node meeting the conditions in  $S$  into  $D$  and removes the node which does not meet the conditions from  $D$  gradually.

**2.2. Related Algorithms.** At present, many local community detection algorithms have been proposed. We introduce two representative local community detection algorithms.

**(1) Clauset Algorithm.** In order to solve the problem of local community detection, Clauset [15] put forward the local

community modularity  $R$  and gave a fast convergence greedy algorithm to find the local community with the greatest modularity.

The definition of local community modularity is as follows:

$$R = \frac{\sum_{ij} A_{ij} \delta(i, j)}{\sum_{ij} A_{ij}}, \quad (1)$$

where  $i$  and  $j$  represent two nodes in the graph. If nodes  $i$  and  $j$  are connected, the value of  $A_{ij}$  is 1; otherwise, it is 0; if nodes  $i$  and  $j$  are both in  $D$ , the value of  $\delta(i, j)$  is 1; otherwise, it is 0.

The local community detection process of Clauset algorithm is similar to that of web crawler algorithm. First, Clauset algorithm starts from an initial node  $v$ . Node  $v$  is added to the subgraph  $D$ , and all its neighbor nodes are added to  $S$ . Then the algorithm adds the node in  $S$  which can bring the maximum increment of  $R$  into the local community iteratively, until the scale of the local community reaches the preset size. That is to say, the algorithm needs to set up a parameter to decide the size of the community, and the result is greatly influenced by the initial node.

**(2) LWP Algorithm.** LWP [16] algorithm is an improved algorithm and it has a clear end condition compared with Clauset algorithm. The algorithm defines another local community modularity  $M$ , which is expressed as

$$M = \frac{(1/2) \sum_{ij} A_{ij} \delta(i, j)}{\sum_{ij} \lambda(i, j)}, \quad (2)$$

where  $i$  and  $j$  represent two nodes in the graph. If nodes  $i$  and  $j$  are connected to each other, the value of  $A_{ij}$  is 1; otherwise, it is 0; if nodes  $i$  and  $j$  are both in  $D$ , the value of  $\delta(i, j)$  is 1; otherwise, it is 0; if only one of the nodes  $i$  and  $j$  is in  $D$ , the value of  $\lambda(i, j)$  is 1; otherwise, it is 0.

Given an undirected and unweighted graph  $G(V, E)$ , LWP algorithm starts from an initial node to find a subgraph with maximum value of  $M$ . If the subgraph is a community (i.e.,  $M > 1$ ), then it returns the subgraph as a community. Otherwise, it is considered that there is no community that can be found starting from this initial node. For an initial node, LWP algorithm finds a subgraph with the maximum value of local modularity  $M$  by two steps. First, the algorithm is initialized by constructing a subgraph with only an initial node  $v$  and all the neighbor nodes of node  $v$  are added to the set  $S$ . Then the algorithm performs incremental step and pruning step.

In the incremental step, the node selected from  $S$  which can make the local modularity of  $D$  increase with the highest value is added to  $D$  iteratively. The greedy algorithm will iteratively add nodes in  $S$  to  $D$ , until no node in  $S$  can be added. In the pruning step, if the local modularity of  $D$  becomes larger when removing a node from  $D$ , then really remove it from  $D$ . In the process of pruning, the algorithm must ensure that the connectivity of  $S$  is not destroyed until no node can be removed. Then update the set  $S$  and repeat the two steps until there is no change in the process. The algorithm has a high Recall, but its accuracy is low.

The complexity of these two algorithms is  $O(K^2d)$ , where  $K$  is the number of nodes to be explored in the local community and  $d$  is the average degree of the nodes to be explored in the local community.

### 3. Description of the Proposed Algorithm

**3.1. Discovery of Minimal Cluster.** Generally, a network can be described by a graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. It contains  $n$  nodes and  $m$  edges.  $C$  represents a node set of a local community in the network and  $|C|$  is the number of nodes in  $C$ . We introduce two definitions related to the algorithm proposed in this paper.

**Definition 1** (neighbor node set). It is a set of nodes connected directly to a single node or a community.

For node  $i$ , its neighbor node set can be expressed as  $N(i) = \{j \mid (i, j) \in E\}$ .

For community  $C$  containing  $n$  nodes, its neighbor node set can be expressed as follows:

$$N(C) = \bigcup_{i=1}^n N(i) - \bigcup_{i=1}^n i. \quad (3)$$

**Definition 2** (number of shared neighbors). The number of shared neighbors for nodes  $i$  and  $j$  can be calculated as

$$W(i, j) = |N(i) \cap N(j)|. \quad (4)$$

The minimal cluster detection is the key of the algorithm. The minimal cluster is the set of nodes that connect to the initial node most closely. We introduce a method proposed in [22] to find the nodes that are closely connected with the initial nodes. It uses the density function  $\Psi$  [23] which is widely used and can be calculated as

$$\Psi(C) = \frac{|C^{\text{in}}|}{\binom{|C|}{2}}, \quad (5)$$

$$\binom{|C|}{2} = \frac{|C|(|C| - 1)}{2},$$

where  $|C^{\text{in}}|$  represents the number of edges in community  $C$  and  $|C|$  represents the number of nodes in community  $C$ . The larger  $\Psi(C)$  is, the more densely the nodes in  $C$  are connected. It is necessary to set a threshold  $\tau(C)$  for  $\Psi(C)$  to decide which nodes are selected to form the initial minimal cluster. Reference [22] gave the definition of this threshold function as shown in

$$\tau(C) = \frac{\sigma(C)}{\binom{|C|}{2}}, \quad \text{where } \sigma(C) = \binom{|C|}{2}^{1-1/\binom{|C|}{2}}. \quad (6)$$

$\tau(C)$  and  $\sigma(C)$  are the thresholds to select the nodes that constitute the minimal cluster  $C$ . If  $\Psi(C) \geq \tau(C)$  or  $|C^{\text{in}}| \geq \sigma(C)$ , these nodes are considered to form a minimal cluster. Compared with other methods, the threshold value does not depend on the artificial setting, but it is totally dependent

```

Input:  $G = (V, E), v$ 
Output: Minimal Cluster  $C$ 
(1)  $C = \Phi$ ;
(2) for  $u \in N(v)$  do
(3)   if  $W(u, v)$  is the largest
(4)     Let  $C = N(u) \cap N(v) \cup \{u, v\}$ ;
(5)   end if
(6) end for
(7) return  $C$ 

```

ALGORITHM 1: Locating minimal cluster.

on the nodes in  $C$ , so the uncertainty of the algorithm is reduced. Through this process, all nodes in the network can be assigned to several densely connected clusters. In the process, the constraint conditions of the minimal clusters are relatively strict. Then the global community structure of the network is found by combining these minimal clusters. This is a process from local to global by finding all minimal clusters to obtain the global structure of the network. Our local community detection algorithm only needs to find one community in the global network. Inspired by this idea, we improve this algorithm as shown in Algorithm 1.

In the network  $G$ , we want to find the minimal cluster containing node  $v$ . First we need to traverse all the neighbors of node  $v$  and to find the node  $u$  which shares the most neighbors with node  $v$  (step 3). Then take nodes  $u, v$  and their shared neighbor nodes as the initial minimal cluster (step 4). Generally speaking, node  $v$  and its neighbor nodes are most likely to belong to the same community. We find the node  $u$  most closely connected with  $v$  according to the number of their shared neighbors. The more the number of their shared neighbors is, the more closely the two nodes are connected. That is to say, the nodes connected with both nodes  $u$  and  $v$  are more likely to belong to the same community. We put them together as the initial minimal cluster of local community expansion, which is effective and reliable verified by experiences.

The process of finding the minimal cluster is illustrated by an example shown in Figure 2. Suppose that we want to find the minimal cluster containing node 1. We need to traverse its neighbor nodes 2, 3, 4, and 6, where  $W(1, 2) = 2$ ,  $W(1, 3) = 3$ ,  $W(1, 4) = 1$ , and  $W(1, 6) = 2$ . We can see that node 3 is the most closely connected one to node 1, so the minimal cluster is  $C = N(1) \cap N(3) \cup \{1, 3\} = \{1, 2, 3, 4, 6\}$ .  $C$  is the starting node set of local community extension.

**3.2. Detection of Local Community.** First of all, we use Algorithm 1 to find the node which is most closely connected to the initial node. We take node  $u$  and node  $v$  as well as their shared neighbor nodes as the initial minimal cluster. The second part of the algorithm is based on the minimal cluster to carry out the expansion of nodes and finally find the local community. The specific process is shown in Algorithm 2.

In the algorithm, we still use  $M$  function used in the LWP algorithm as the criteria of local community expansion. Algorithm 1 can find the initial minimal cluster  $C$ . After that,

Input:  $G = (V, E), C$   
 Output: Local Community LC  
 (01) Let  $LC = C$   
 (02) Calculate  $N(LC), M$   
 (03) While  $N(LC) \neq \Phi$  do  
 (04)   foreach  $u \in N(LC)$   
 (05)     if  $\Delta M$  is the largest  
 (06)        Let  $LC = LC \cup \{u\}$   
 (07)     End if  
 (08)   End for  
 (09)   Update  $N(LC), M$   
 (10) Until no node can be added into LC  
 (11) Return LC

ALGORITHM 2: Local community detection.

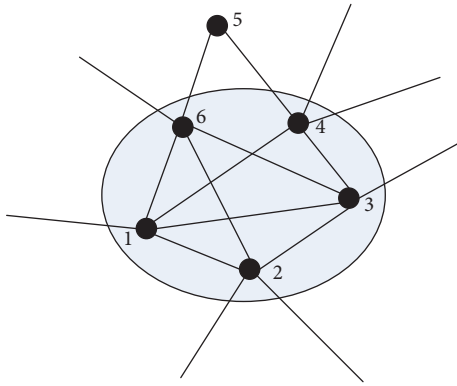


FIGURE 2: The discovery of minimal cluster.

Algorithm 2 finds the neighbor node set  $N(LC)$  of LC and calculates the initial value of  $M$  (step 02). Then it traverses all the nodes in  $N(LC)$  (steps 03-04) to find a node which can make  $\Delta M$  maximum and add it into the local community LC (steps 05-08); update  $N(LC)$  and  $M$  (step 09) until no new node is added to LC (step 10).

The complexity of the NewLCD algorithm is almost the same as the Clauset algorithm. The NewLCD algorithm uses extra time of finding minimal cluster which is linear to the degree of the initial node  $v$ .

## 4. Experimental Results and Analysis

In this section, the NewLCD algorithm is compared with several representative local community detection algorithms, namely, LWP, LS, and Clauset, to verify its performance. The experimental environment is the following: Intel (R) Core (TM) i5-2400 CPU @ 3.10 GHz; memory 2 G; operating system: Windows 7; programming language: C#.Net.

**4.1. Experimental Data.** The dataset of LFR benchmark networks and three real network datasets are used in the experiments.

(1) LFR benchmark networks [24] are currently the most commonly used synthetic networks in community detection.

TABLE 1: LFR benchmark network information.

Network ID	$N$	$k$	$\max k$	$\min c$	$\max c$	$\mu$
B1	1000	20	50	10	50	0.1~0.9
B2	1000	20	50	20	100	0.1~0.9
B3	5000	20	50	10	50	0.1~0.9
B4	5000	20	50	20	100	0.1~0.9

TABLE 2: Real network information.

Network ID	Name	Number of nodes	Number of edges	Reference
R1	Karate	34	78	[19, 20]
R2	Football	115	613	[19, 21]
R3	Polbooks	105	441	[19]

It includes the following parameters:  $N$  is the number of nodes;  $\min c$  is the number of nodes that the minimum community contains;  $\max c$  is the number of nodes that the biggest community contains;  $k$  is the average degree of nodes in the network;  $\max k$  is the maximum degree of node;  $\mu$  is a mixed parameter, which is the probability of nodes connected with nodes of external community. The greater  $\mu$  is, the more difficult it is to detect the community structure. We generate four groups of LFR benchmark networks. Two groups of networks, B1 and B2, share the common parameters of  $N = 1000$ ,  $k = 20$ , and  $\max k = 50$ . The other two groups of networks, B3 and B4, share the common parameters of  $N = 5000$ ,  $k = 20$ , and  $\max k = 50$ . The community size  $\{\min c, \max c\}$  of B1 and B3 is  $\{10, 50\}$  and the community size  $\{\min c, \max c\}$  of B2 and B4 is  $\{20, 100\}$ , implying small community networks and large community networks, respectively; each group contains nine networks with  $\mu$  ranging from 0.1 to 0.9 representing from low to high hybrid network. The details are shown in Table 1.

(2) We choose three real networks including Zachary's Karate club network (Karate), American college Football network (Football), and American political books network (Polbooks). The detailed information is shown in Table 2.

**4.2. Experiments on Artificial Networks.** Because of the large size of the synthetic networks, 50 representative nodes are randomly selected from each group as the initial node and all the experimental results are averaged as the final result. Figures 3-6 are the comparison chart of the experimental results of each algorithm on the four groups of LFR benchmark networks (B1-B4). The ordinate represents the three evaluation criteria for local community detection, respectively, and the abscissa is the value of  $\mu$  (0.1-0.9). The following conclusions can be obtained by observation.

(1) LS and LWP algorithms have higher Precision compared with Clauset algorithm. But their Recall value is lower than Clauset algorithm. LS and LWP algorithms cannot have both high accuracy and Recall. Their comprehensive effect may be not higher than the benchmark algorithm Clauset.

(2) All these three indicators of NewLCD algorithm are significantly higher than Clauset algorithm, which shows that the initial state indeed affects the results of local community

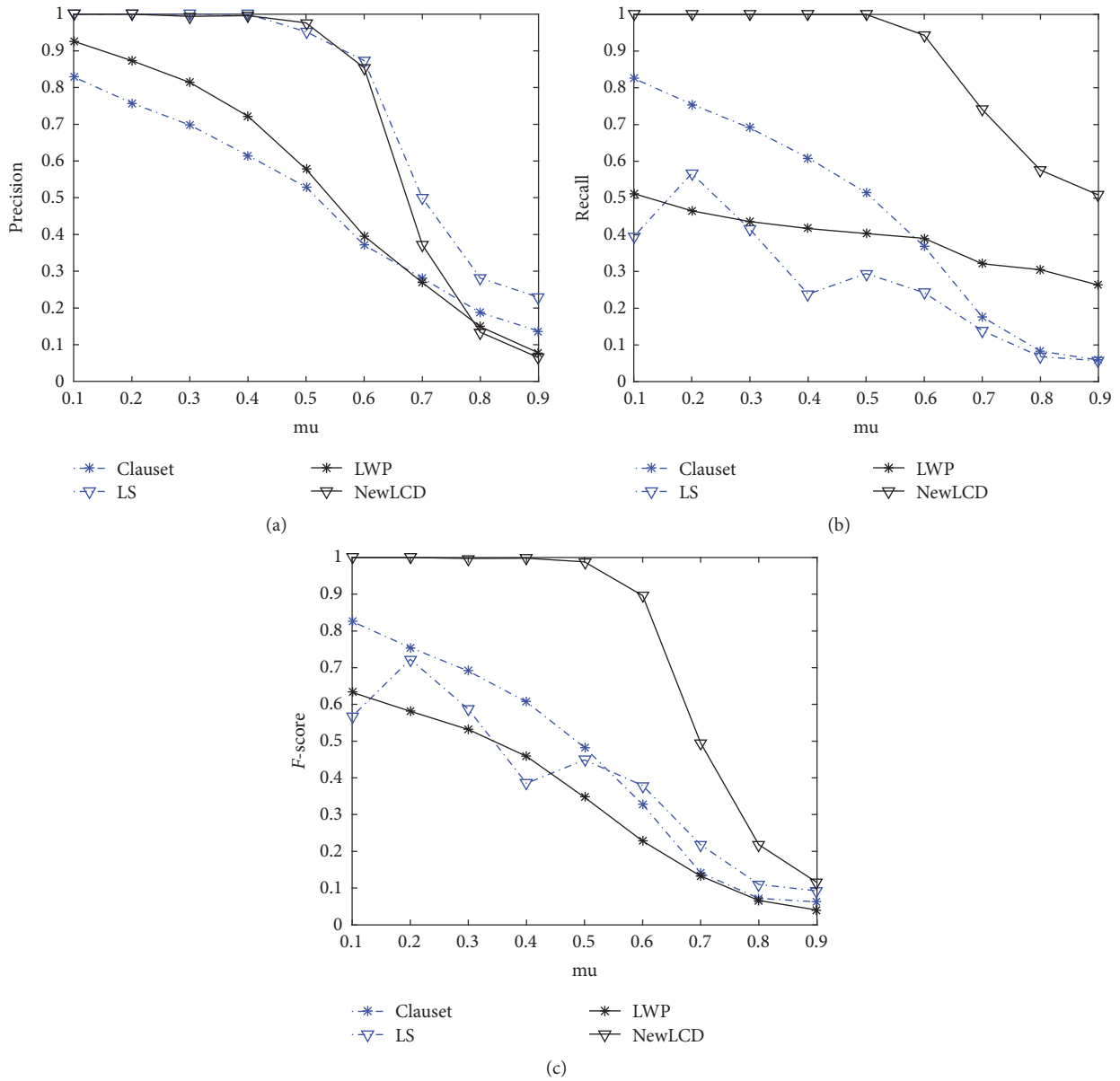


FIGURE 3: Comparison of BI.

detection algorithm, and starting from the minimal cluster is better than a single node.

(3) Overall, NewLCD algorithm is the best. On the four groups of networks, when the parameter  $\mu$  is less than 0.5, NewLCD algorithm can find almost all the local communities where each node is located. In high hybrid networks, when the value of  $\mu$  is greater than 0.8, the local community detection effect of NewLCD algorithm is not good, just like other algorithms. The main reason is that the community structure of the network is not obvious.

In summary, NewLCD algorithm can detect better local communities on the artificial networks than the other three local community detection algorithms.

**4.3. Experiments on Real Networks.** In order to further verify the effectiveness of NewLCD algorithm, we compare

it with three other algorithms on three real networks (Karate, Football, and Polbooks). These three networks are often used to verify the effectiveness of algorithms on complex networks. The experimental results are shown in Table 3 and the maximum values of each indicator are presented in boldface. The maximum value of Precision on Karate is 0.989 obtained by LS algorithm. But its Recall value is just 0.329 which is the minimum value among these four algorithms. So the result of LS algorithm is the worst. On Karate networks, Clauset algorithm and LWP algorithm have the same problem as LS, which means that their Recall value is low. While the Recall and F-score values of NewLCD algorithm are the largest, NewLCD algorithm is optimal. On the Football network, the comprehensive effect of NewLCD algorithm is also the best. On the Polbooks network, the advantages of NewLCD algorithm are more obvious, and the three indicators of its

TABLE 3: The comparison of algorithms on the real networks.

Dataset	Evaluation criteria	Clauset	LS	LWP	NewLCD
Karate	Precision	0.927	<b>0.989</b>	0.884	0.934
	Recall	0.526	0.329	0.529	<b>0.809</b>
	<i>F</i> -score	0.671	0.494	0.662	<b>0.867</b>
Football	Precision	0.803	<b>0.943</b>	0.680	0.880
	Recall	0.878	0.732	0.712	<b>0.940</b>
	<i>F</i> -score	0.839	0.824	0.696	<b>0.909</b>
Polbooks	Precision	0.741	0.879	0.770	<b>0.914</b>
	Recall	0.442	0.182	0.477	<b>0.757</b>
	<i>F</i> -score	0.554	0.301	0.589	<b>0.828</b>

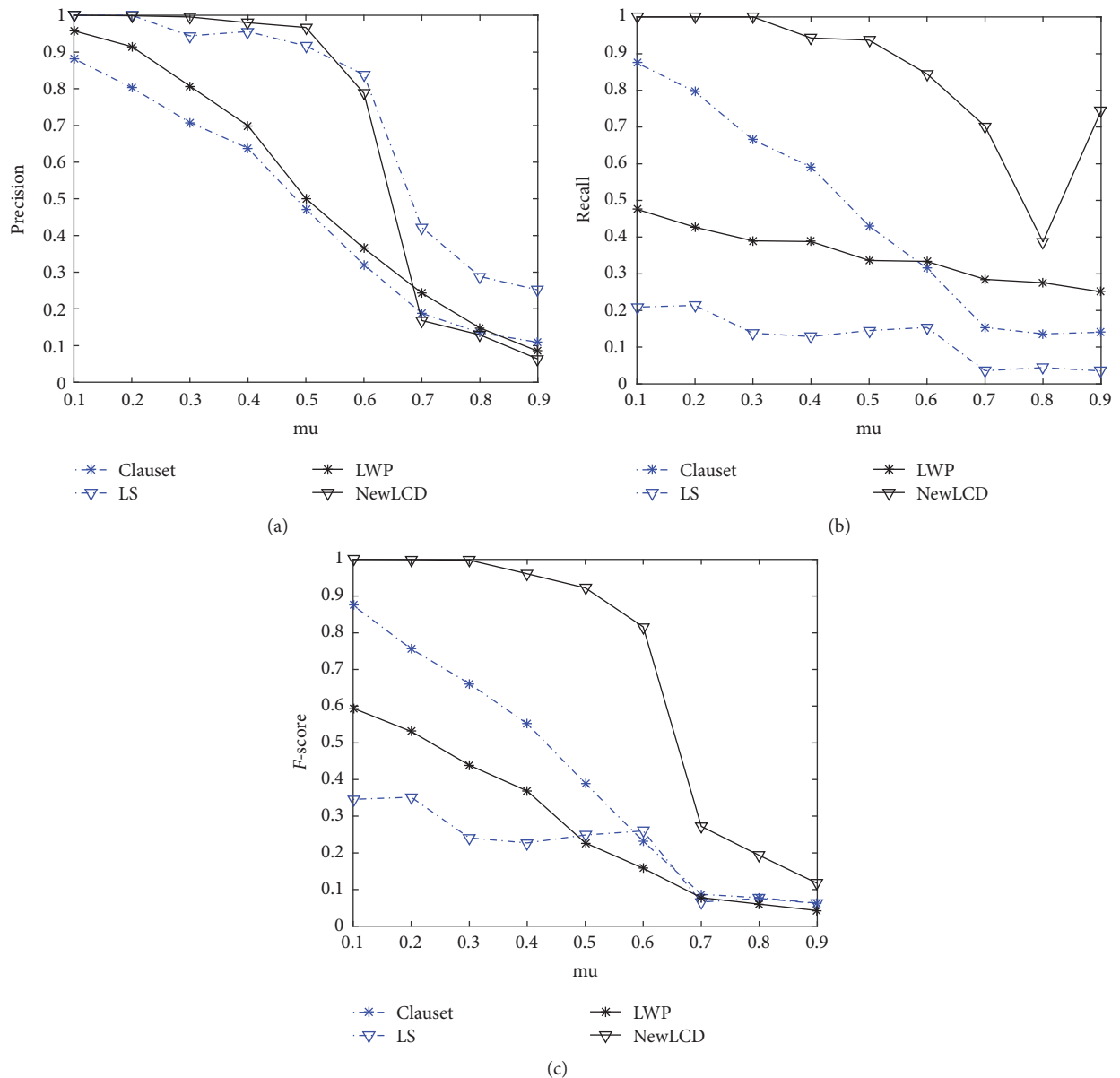


FIGURE 4: Comparison of B2.

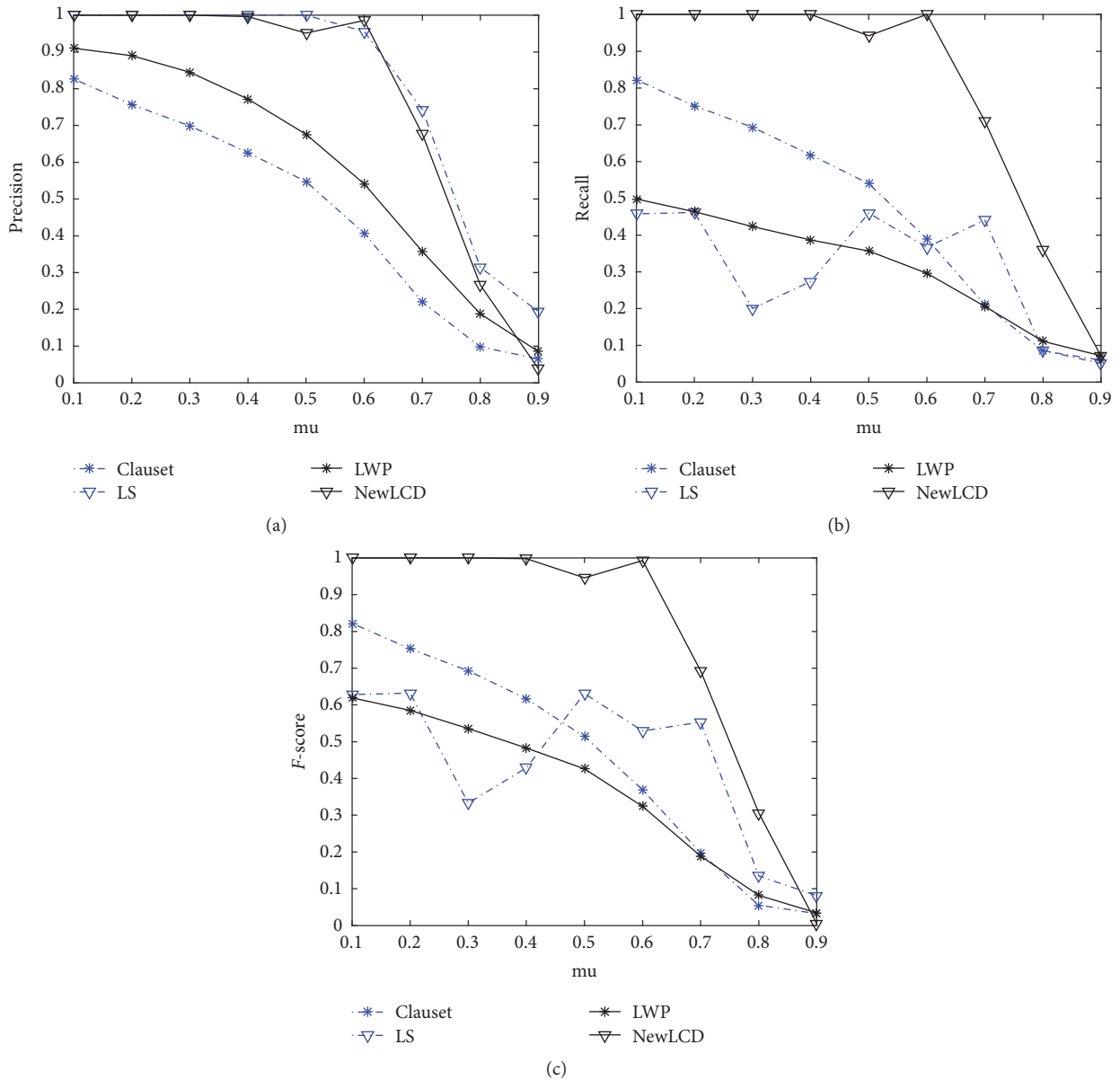


FIGURE 5: Comparison of B3.

results are all the best. In summary, not only can NewLCD algorithm be effectively applied on the artificial network, but it can also be very effective on the real networks.

Karate network is a classic interpersonal relationship network of sociology. It reflects the relationship between managers and trainees in the club. The network is from a Karate club in an American university. The club's administrator and instructor have different opinions on whether to raise the club fee. As a result, the club splits into two independent small clubs. Since the structure of Karate network is simple and it reflects the real world, many community detection algorithms use it as the standard experimental dataset to verify the quality of the community. In order to further verify the effectiveness of the algorithm,

we do a further experiment on Karate. Figure 7 is the real community structure of Karate. If we select node 8 as the initial node, Figures 8 and 9 are, respectively, the local community structure detected by NewLCD and Clauset.  $\{9, 14, 15, 18, 20, 22, 28, 27, 24, 25, 32, 23, 26, 29, 8, 30, 33, 31\}$  is the real local community containing node 8 and  $\{14, 15, 18, 20, 22, 26, 29, 9, 8, 30, 32, 33, 23, 27, 2, 28\}$  is the result of Clauset. We can see that node 2 is assigned to the local community, while nodes 23, 24, 25, and 31 are left out. The community containing node 8 detected by NewLCD is  $\{9, 14, 15, 18, 20, 22, 28, 27, 24, 25, 32, 23, 26, 29, 8, 2, 30, 33, 31\}$ . Only node 2 is wrongly assigned to the community and there is no omission of any node. The local community detected by NewLCD is more similar to the real one. While a node cannot represent

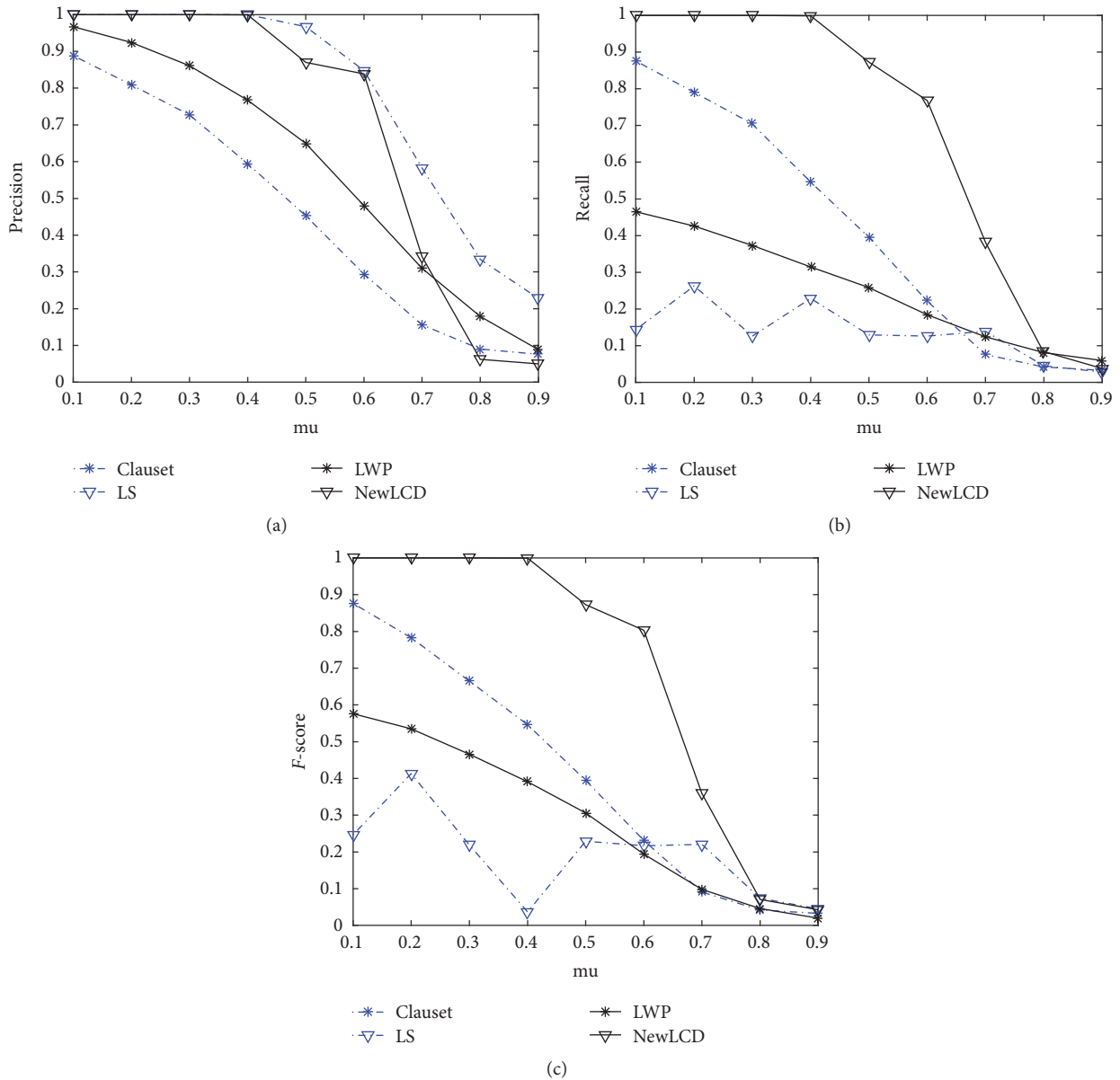


FIGURE 6: Comparison of B4.

all situation, we do more experiments expanding from each node of Karate and compare the corresponding Precision, Recall, and  $F$ -score, as shown in Figure 10. The abscissa represents the 34 nodes, from 0 to 33, and the ordinate is  $F$ -score, Recall, and Precision, respectively. Although the Precision values of Clauset are slightly higher than the results of NewLCD expanding from nodes {4, 5, 6, 10, 23, 26, 27, 29}, the Recall values of Clauset are far lower than the results of NewLCD. So NewLCD algorithm is much better than Clauset algorithm.

## 5. Conclusion

This paper proposes a new local community detection algorithm based on minimal cluster—NewLCD. This algorithm

mainly consists of two parts. The first part is to find the initial minimal cluster for local community expansion. The second part is to add nodes from the neighbor node set which meet the local community condition into the local community. We compare the improved algorithm with other three local community detection algorithms on the real and artificial networks. The experimental results show that the proposed algorithm can find the local community structure more effectively than other algorithms.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper. They declare that they do not have any commercial or associative interest that



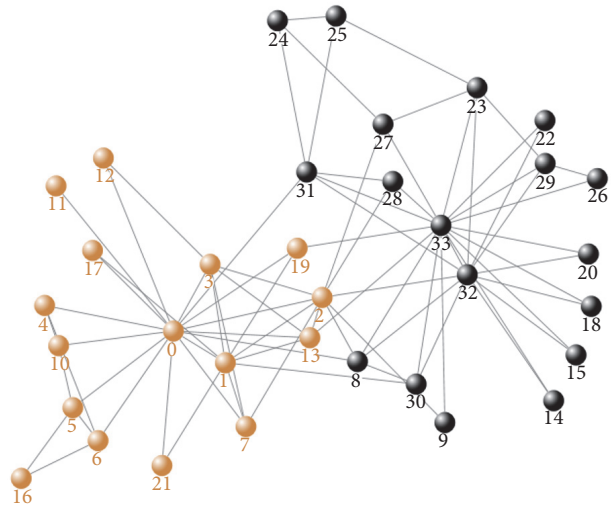


FIGURE 7: The real community structure of Karate.

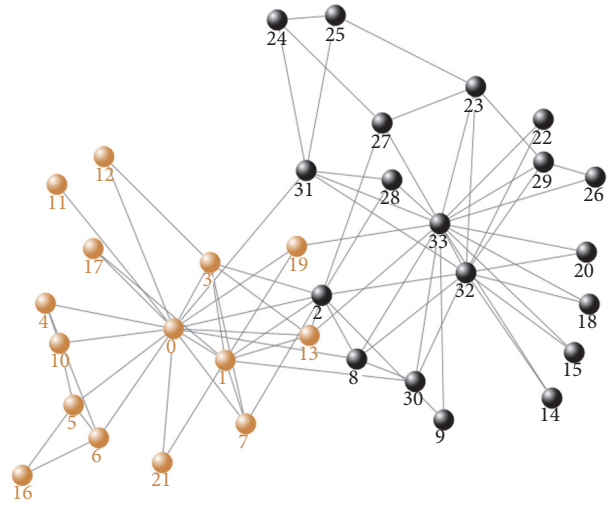


FIGURE 8: The result of NewLCD algorithm on Karate.

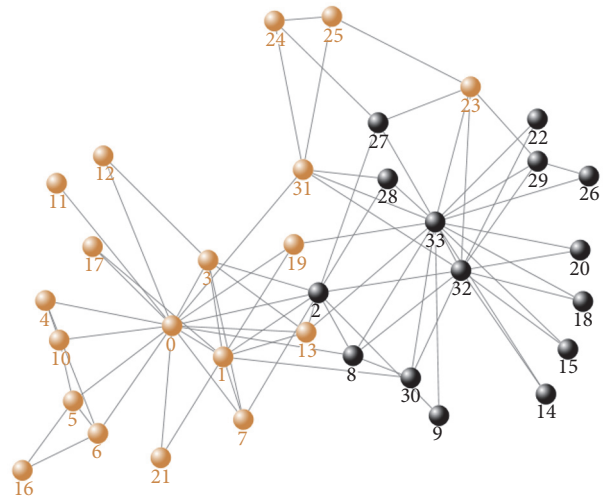


FIGURE 9: The result of Clauset algorithm on Karate.

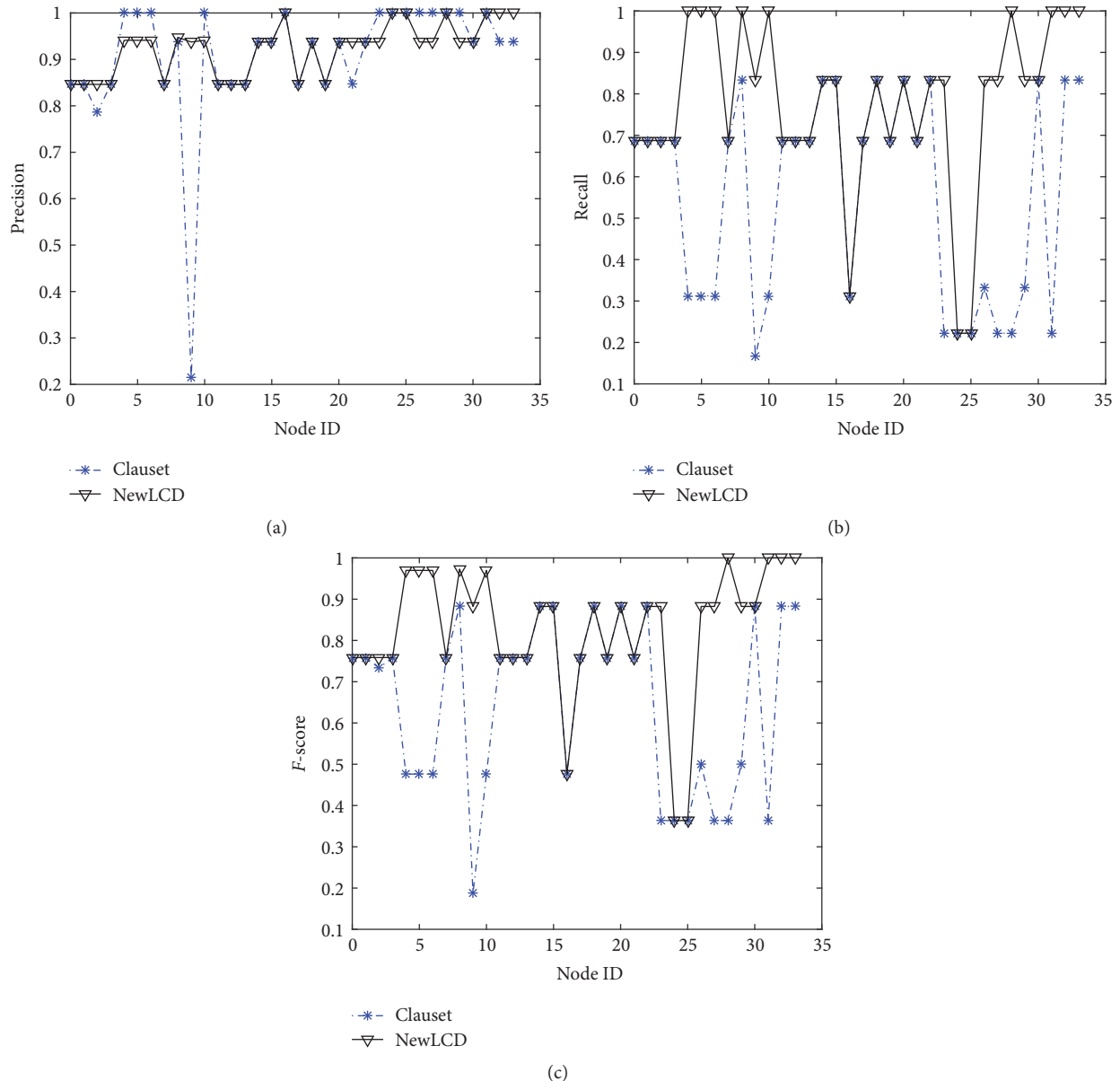


FIGURE 10: The contrast results of each node in Karate.

represents a conflict of interests in connection with the work submitted.

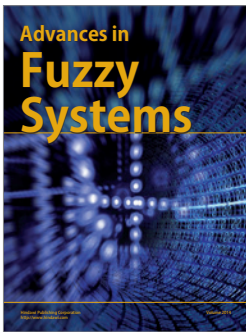
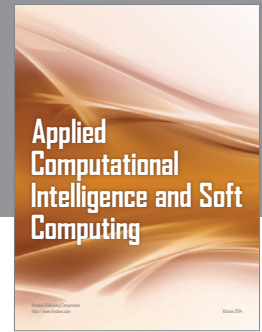
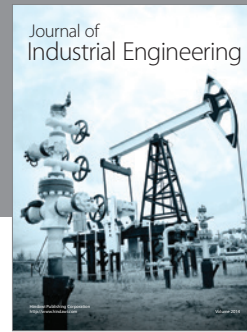
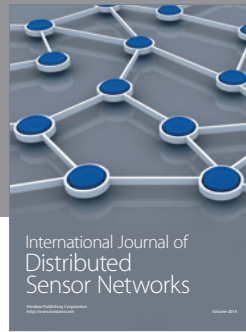
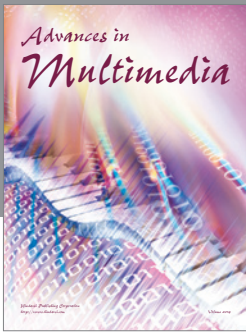
## Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61572505, no. 51404258, and no. 61402482), the National High Technology Research and Development Program of China (no. 2012AA011004), China Postdoctoral Science Foundation (no. 2015T80555), and Jiangsu Planned Projects for Postdoctoral Research Funds (no. 1501012A).

## References

- [1] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E-Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, pp. 292–313, 2004.
- [2] J. Lee, S. P. Gross, and J. Lee, "Modularity optimization by conformational space annealing," *Physical Review E*, vol. 85, no. 5, Article ID 056702, pp. 499–508, 2012.
- [3] H.-W. Shen and X.-Q. Cheng, "Spectral methods for the detection of network community structure: a comparative analysis," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2010, no. 10, Article ID P10020, 2010.
- [4] J. Wu, Z.-M. Cui, Y.-J. Shi, S.-L. Sheng, and S.-R. Gong, "Local density-based similarity matrix construction for spectral clustering," *Journal on Communications*, vol. 34, no. 3, pp. 14–22, 2013.
- [5] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. A. K. Suykens, "Multiclass semisupervised learning based upon kernel spectral clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 720–733, 2015.

- [6] K. Taşdemir, B. Yalçın, and I. Yildirim, “Approximate spectral clustering with utilized similarity information using geodesic based hybrid distance measures,” *Pattern Recognition*, vol. 48, no. 4, pp. 1461–1473, 2015.
- [7] V. D. Blondel, J. Guillaume, R. Lambiotte et al., “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 30, no. 2, pp. 155–168, 2008.
- [8] K. M. Tan, D. Witten, and A. Shojaie, “The cluster graphical lasso for improved estimation of Gaussian graphical models,” *Computational Statistics and Data Analysis*, vol. 85, pp. 23–36, 2015.
- [9] F. De Morsier, D. Tuia, M. Borgeaud, V. Gass, and J.-P. Thiran, “Cluster validity measure and merging system for hierarchical clustering considering outliers,” *Pattern Recognition*, vol. 48, no. 4, pp. 1478–1489, 2015.
- [10] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, “Efficient agglomerative hierarchical clustering,” *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.
- [11] L. Subelj and M. Bajec, “Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction,” *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, vol. 83, no. 3, pp. 885–896, 2011.
- [12] S. Li, H. Lou, W. Jiang, and J. Tang, “Detecting community structure via synchronous label propagation,” *Neurocomputing*, vol. 151, no. 3, pp. 1063–1075, 2015.
- [13] Y. Yi, Y. Shi, H. Zhang, J. Wang, and J. Kong, “Label propagation based semi-supervised non-negative matrix factorization for feature extraction,” *Neurocomputing*, vol. 149, pp. 1021–1037, 2015.
- [14] D. Zikic, B. Glocker, and A. Criminisi, “Encoding atlases by randomized classification forests for efficient multi-atlas label propagation,” *Medical Image Analysis*, vol. 18, no. 8, pp. 1262–1273, 2014.
- [15] A. Clauset, “Finding local community structure in networks,” *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 72, no. 2, pp. 254–271, 2005.
- [16] F. Luo, J. Z. Wang, and E. Promislow, “Exploring local community structures in large networks,” *Web Intelligence and Agent Systems*, vol. 6, no. 4, pp. 387–400, 2008.
- [17] Y. J. Wu, H. Huang, Z. F. Hao, and F. Chen, “Local community detection using link similarity,” *Journal of Computer Science and Technology*, vol. 27, no. 6, pp. 1261–1268, 2012.
- [18] Q. Chen, T.-T. Wu, and M. Fang, “Detecting local community structures in complex networks based on local degree central nodes,” *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 3, pp. 529–537, 2013.
- [19] <http://www-personal.umich.edu/~mejn/netdata/>.
- [20] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
- [21] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [22] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, “Overlapping communities in dynamic networks: their detection and mobile applications,” in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom '11)*, pp. 85–95, Las Vegas, Nev, USA, September 2011.
- [23] S. Fortunato and C. Castellano, “Community structure in graphs,” in *Computational Complexity*, pp. 490–512, Springer, 2012.
- [24] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Physical Review E*, vol. 78, no. 4, Article ID 046110, pp. 561–570, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

