

## Research Article

# Local Community Detection in Complex Networks Based on Maximum Cliques Extension

**Meng Fanrong, Zhu Mu, Zhou Yong, and Zhou Ranran**

*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Zhu Mu; [zhumubest@163.com](mailto:zhumubest@163.com)

Received 6 January 2014; Accepted 25 February 2014; Published 15 April 2014

Academic Editor: Guanghui Wen

Copyright © 2014 Meng Fanrong 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.

Detecting local community structure in complex networks is an appealing problem that has attracted increasing attention in various domains. However, most of the current local community detection algorithms, on one hand, are influenced by the state of the source node and, on the other hand, cannot effectively identify the multiple communities linked with the overlapping nodes. We proposed a novel local community detection algorithm based on maximum clique extension called LCD-MC. The proposed method firstly finds the set of all the maximum cliques containing the source node and initializes them as the starting local communities; then, it extends each unclassified local community by greedy optimization until a certain objective is satisfied; finally, the expected local communities will be obtained until all maximum cliques are assigned into a community. An empirical evaluation using both synthetic and real datasets demonstrates that our algorithm has a superior performance to some of the state-of-the-art approaches.

## 1. Introduction

In recent years, more and more research has begun to pay attention to large complex networks, such as social networks, protein interaction networks, citation networks, and WWW. Extensive researches have indicated that community structure universally exists in complex networks and the connection between nodes in a community is closer than that between communities. Meanwhile, these nodes often have similar attributes or play a similar role. Therefore, community detection has become one of the basic tasks of complex network analysis and is of important theoretical significance and real value.

The research of community detection mostly has focused on detecting all community structures in a whole network from a global viewpoint [1–5]. However, the large scale of complex networks in many real applications is inconceivable. For example, the friend relation networks on Facebook and Twitter contain hundreds of millions of nodes [6], and the detection of the community structures in such huge complex networks will cost tremendous time and space. In addition, as the nodes and links of many complex networks are dynamically evolving [7, 8], it is often hard for us to acquire the complete network information, further increasing the

difficulty in global community detection. Therefore, many scholars have begun to focus on local community detection of complex networks.

Different from the global community detection which classifies a total complex network, local community detection is only to inquire the community structure where a designated node (source node) is located in a network. A complex network is essentially divided into two parts, namely, the community where a designated node is located and the rest part. Furthermore, the local community where the node is located has a close internal connection within the community but a relatively loose relation with the outside. Local community detection need not know all information about a complex network in advance. It starts from a node, gradually extends from the node, and gradually acquires the local information around the current community during the extension process. The representative algorithms for local community detection include [9–13].

However, most of available local community detection algorithms have two restrictions: firstly, the method including direct start from a source node, continuous selection of the best nodes from candidate ones by greedy algorithm, and adding them into a local community till the local community detected satisfies all specified conditions, which makes it

easy to deviate from the real local community, thus reducing the accuracy of local community detection; secondly, in this way, finally, only a unique local community structure can be obtained and when the source node is an overlapped (hub) node connecting multiple communities, it is unable to obtain all local communities.

To solve the above two problems, we propose a local community detection algorithm based on maximum cliques extension (LCD-MC), which includes finding maximum cliques and extending local communities. Its main advantages are shown as below.

- (i) Instead of taking the source node as input directly, finding all maximum cliques containing the source node is made as the start of local community extension, thus increasing the stability of local community detection.
- (ii) The flexibility in identifying overlapped node-involved local communities is realized by extending all maximum cliques satisfying certain conditions, respectively.
- (iii) The experimental results on both synthetic and real networks demonstrate that, compared with the state-of-the-art local community detection algorithms, LCD-MC, on one side, can obtain better local community quality and, on the other side, can effectively identify multiple local community structures connected with the overlapped node.

## 2. Related Work

Let a complex network  $G = (V, E)$ , where  $V$  represents the node set,  $E$  the edge set, and  $n$  and  $m$  number of nodes and number of edges in the network, respectively. Different from global algorithms which divide  $G$  into a number of closely connected community structures, local community detection designates a node  $v_0$  and explores the community structures in close relation with the node  $v_0$ .

Clauset firstly proposed the formal definition of local community detection [9]. Assume that we have known a community structure  $C$  (initially,  $C$  contains only the node  $v_0$ ) composed of some nodes; set  $U$  is connected with nodes in  $C$  but does not belong to the node set of community  $C$ . The process of local community detection is to continuously select nodes from  $U$  and add them to the current community  $C$  till the predefined local modularity  $R$  reaches the maximum value. To define the objective function  $R$ , Clauset also defined the boundary  $B$  of community  $C$ , that is, node sets in  $C$  that have at least one node connecting with  $U$ , as shown in Figure 1.

Assume that the given  $C$ ,  $B$ , and  $U$  are known;  $R$  is defined by

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

where  $B_{ij} = 1$  if  $(i, j) \in E$  and either node  $v_i$  or node  $v_j$  exists in  $B$ ; otherwise,  $B_{ij} = 0$ . When either node  $v_i$  or node  $v_j$  exists

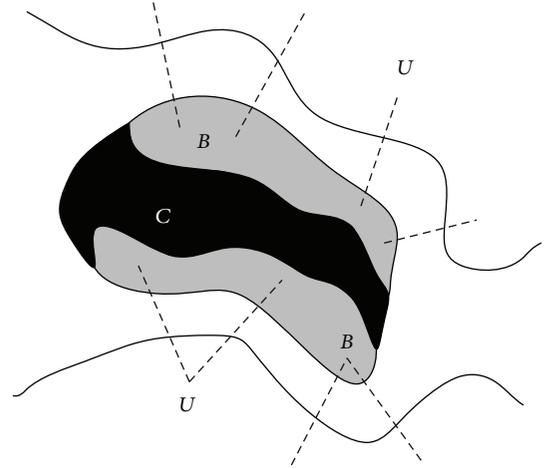


FIGURE 1: Local community detection model.

in  $B$  and the other node exists in  $C$ , then  $\delta(i, j) = 1$  and, otherwise,  $\delta(i, j) = 0$ . The larger the  $R$  value is, the better the local community structure detected will be. Initially,  $C = \{v_0\}$ , and Clauset used a greedy optimization algorithm for the local modularity  $R$  to find the local community structure where the designated  $v_0$  is located.

Similar to Clauset algorithm, Luo et al. proposed LWP algorithm [10], in which  $R$  is replaced by a new local modularity  $M$ , as shown in the following:

$$M = \frac{M_{\text{in}}}{M_{\text{out}}} = \frac{(1/2) \sum_{ij} A_{ij} \theta(i, j)}{\sum_{ij} A_{ij} \lambda(i, j)}, \quad (2)$$

where  $\theta(i, j) = 1$  if both node  $v_i$  and node  $v_j$  exist in community  $C$ ; otherwise,  $\theta(i, j) = 0$ . And  $\lambda(i, j) = 1$  means that only one, either node  $v_i$  or node  $v_j$ , exists in community  $C$ . In addition to a different objective function, LWP algorithm also includes addition and deletion operations, making it possible to add into or delete from community  $C$  the nodes that can increase  $M$  value. Besides, LWP algorithm need not predefine the size of community  $C$  in advance.

LMF [11] is a local-global community detection algorithm, which proposes a fitness function, as shown in the following:

$$F = \frac{k_{\text{in}}^C}{(k_{\text{in}}^C + k_{\text{ex}}^C)^\alpha}, \quad (3)$$

where  $k_{\text{in}}^C$  and  $k_{\text{ex}}^C$  refer to the sums of the internal node degrees and external node degrees of community  $C$ , respectively, and  $\alpha$  is a resolution parameter used for controlling the size of local community. This algorithm is similar to LWP algorithm in that, according to the given  $\alpha$ , it achieves the objective of making the fitness function  $F$  reach the maximum local value by addition and deletion.

Wu et al. [12] proposed a local community detection algorithm based on link similarity (LS). The algorithm firstly defines the similarity between a single node and a local community and then carries out local community detection

in a decrease sequence of the calculated similarity values. In addition, this algorithm's search process is composed of greedy clustering, optimization, and trimming.

Chen et al. [13] proposed a local community detection algorithm based on local degree center node (LMD). Though the objective of local community detection is to find the community structure where the given node is located, it was held by the authors that, for some given nodes, the detection directly starting from  $v_0$  may not necessarily obtain ideal results. Therefore, to increase the robustness of local community detection, instead of starting the search from the given node, LMD starts from finding a local degree center node nearest to the given node and then extends local communities starting from this local degree center node with  $R$ ,  $M$ , and  $F$  as objective functions. Here, the degree of the local degree center node is greater than or equal to that of all neighbor nodes.

### 3. Algorithm

In this section, we introduce the proposed local community detection algorithm based on maximum cliques extension (LCD-MC), which is mainly composed of two parts, namely, algorithm FindMC for finding the maximum cliques of a node and algorithm LCD for local community extension corresponding to the maximum cliques.

#### 3.1. FindMC Algorithm for Finding Maximum Cliques

*Definition 1.* Given an undirected graph  $G = (V, E)$ , if  $U \subseteq V$ , for random  $u, v \subseteq U$ , and  $(u, v) \subseteq E$ , then  $U$  is called  $G$ 's complete subgraph.

*Definition 2.*  $G$ 's complete subgraph  $U$  is  $G$ 's maximum clique if and only if  $U$  is not included in the  $G$ 's larger complete subgraphs.

In FindMC, we mainly adopt the concept of the Bron-Kerbosch algorithm [19] to find the maximum cliques where the given source node  $v_0$  is located. That means mainly using three sets, namely,  $R$ ,  $P$ , and  $X$ , whose functions are, respectively, explained as follows:

- (1)  $R$  is used for storing the already acquired nodes forming the current clique structures;
- (2)  $P$  is used for storing all the candidate nodes edge-connected with the nodes in  $R$ , which can be used to extend all the clique structures already found and will be added to  $R$ ;
- (3)  $X$  is used for storing the candidate nodes that have been used in  $P$ . If a certain node in  $P$  can cluster with nodes in  $R$  to form a larger clique structure, this node will be added into both set  $R$  and set  $X$  but deleted from set  $P$ .

FindMC algorithm starts from node  $v_0$  and constructs recursively a search tree as the nodes in set  $P$  are continuously added to set  $R$ . On this search tree, each internal node corresponds to a state or is called a candidate clique structure,

while a leaf node represents a corresponding maximum clique.

Algorithm 1 gives the pseudocode of FindMC algorithm for finding maximum clique. In the initialization phase, FindMC stores only node  $v_0$  in set  $R$  and only  $v_0$ 's neighbours in set  $P$ . This is because the nodes in the maximum clique where node  $v_0$  is located can only be  $v_0$ 's neighbours and any search outside this will be invalid (line 01). In addition, to protect the updating of set  $P$ , the nodes in  $P$  are copied to  $nodeList$  to be extended, where the nodes are arranged in a degree-decreasing sequence. This is because that node with a larger degree is easier to form a maximum clique. Therefore, they are given a priority so as to improve the algorithm efficiency (lines 02~03). After the initialization is completed, FindMC algorithm, at its second phase, will execute the conventional Bron-Kerbosch algorithm recursively according to sets  $R$ ,  $P$ , and  $X$  and store the acquired maximum cliques in MCS (maximum clique set).

Algorithm 2 is the conventional Bron-Kerbosch algorithm. In this algorithm, if both sets  $P$  and  $X$  are null sets, then the nodes in  $R$  satisfy the condition for maximum clique and will be added to the maximum clique set MCS (lines 01~03). Otherwise, it will select a pivot node  $u$  from sets  $P$  and  $X$  and continue to make self-recursive call for each node  $v$  except  $u$  in  $P$ . When all subprograms of  $v$  are ended,  $v$  will be deleted from set  $P$  and added into set  $X$  (lines 04~09).

#### 3.2. LCD Algorithm for Extending Local Community Structure.

A maximum clique is a very closely connected node set; however, the requirement on full connection is too strict to the definition of community structure. Therefore, the second step of LCD-MC is to further extend the local community structure according to the MCS obtained in Algorithm 1, as shown in Algorithm 3 (LCD). LCD algorithm carries out the following operation for each unclassified maximum clique (no node in such clique has been allocated to any local community):

- (1) initialization, which will add each node in the current clique to the local community LC and add all nodes connected with but not belonging to LC into set  $U$  (lines 04~08);
- (2) extending local community LC, which will select a node  $t$  from  $U$  that can make the greatest increase of the objective function, add it to the local community LC, and update the corresponding nodes in  $U$  till there is not any node that can increase the value of the objective value (lines 09~21);
- (3) finally, return the required local community set LCS, each community of which containing the initial node  $v_0$  (line 23).

On line 12 of Algorithm 3, function CalculateDeltaValue ( $u$ ) calculates the incremental value of the objective function after node  $u$  is added to the current local community. Here, the objective function can be any objective function mentioned in Section 2. To improve the algorithm efficiency, the incremental value of objective function can be calculated

```

Input:  $G = (V, E), v_0$ 
Output: Maximum Clique Set MCS
Begin:
(1)  $R \leftarrow \{v_0\}, P \leftarrow \{w \mid (v_0, w) \in E\}, X \leftarrow \emptyset, MCS \leftarrow \emptyset$ 
(2)  $nodeList \leftarrow P$ 
(3) sort the nodes in  $nodeList$  based on their degree descending
(4) foreach  $v$  in  $nodeList$ 
(5)   Bron_Kerbosch ( $R \cup \{v\}, P \cap N(v), X \cap N(v), MCS$ );
(6)    $P \leftarrow P \setminus \{v\}$ ;
(7)    $X \leftarrow X \cup \{v\}$ ;
(8) endfor
(9) retrun MCS
End

```

ALGORITHM 1: FindMC.

```

Input:  $R, P, X, MCS$ 
Output: MCS
Begin:
(1) if  $P = \emptyset$  and  $X = \emptyset$ 
(2)    $MCS \leftarrow MCS \cup R$ ;
(3) else
(4)    $u \leftarrow$  a pivot node in  $P \cup X$ 
(5)   foreach  $v$  in  $P \setminus \{u\}$ 
(6)     BronKerbosch ( $R \cup \{v\}, P \cap N(v), X \cap N(v), MCS$ );
(7)      $P \leftarrow P \setminus \{v\}$ ;
(8)      $X \leftarrow X \cup \{v\}$ ;
(9)   endfor
(10) endif
(11) retrun MCS
End

```

ALGORITHM 2: Bron-Kerbosch.

according to the assumed change value generated after the addition of node  $u$  into the local community. Take the objective function  $M$  in (2) as an example. Assume the number of edges in the current local community LC is  $E_{in}$ , the number of the edges between LC and  $U$  is  $E_{out}$ , where  $x$  represents the delta value of edge number in LC due to the addition of  $u$ , and  $y$  represents the number of edges of nodes in LC connecting  $U$  after the addition of  $u$  into LC; then, the delta value of  $M$  is

$$\text{delta} = \frac{x E_{out} + (x - y) E_{in}}{(E_{out} - x + y) E_{in}}. \quad (4)$$

**3.3. Time Complexity.** Let us analyze the time complexity of LCD-MC from its two steps, respectively. As the worst time complexity of Bron-Kerbosch algorithm is  $O(3^{n/3})$  [20], but LCD-MC only needs to find cliques containing the initial node, it essentially operates Bron-Kerbosch algorithm in the subgraphs formed by the initial node and its neighbors. In this way, the worst time complexity of FindMC, the first step in LCD-MC, is  $O(3^{k/3})$ , where  $k$  is the degree of the initial node,

and that of LCD, the second step in LCD-MC, is  $O(|C|^2)$ , where  $C$  is the local community detected. It should be noted that either  $k$  in FindMC or  $|C|$  in LCD is far smaller than the overall size of the network. Therefore, LCD-MC indicates satisfied time efficiency.

## 4. Evaluation

In this section, we make the verification comparison of LCD-MC proposed in this paper with several representative local community detection algorithms. We conduct all the experiments on a Pentium Core2 Duo 2.8 GHz PC with 2 GBytes of main memory, running on Windows 7. We implement our algorithm in C#, using Microsoft Visual Studio 2008.

**4.1. Experimental Setup.** We compared LCD-MC with several representative local community detection algorithms including Clauset [9], LWP [10], LS [12], and LMD [13]. Of them, Clauset was the first to propose local community detection, and, therefore, we take it as the basic algorithm. Both

```

Input:  $G = (V, E)$ , MCS
Output: Local Communities LCS
Begin:
(1)  $LCS \leftarrow \emptyset, U \leftarrow \emptyset$ 
(2) foreach unclassified MC in MCS
(4)  $LC \leftarrow \emptyset, U \leftarrow \emptyset$ 
(5) foreach  $v$  in MC
(6)  $LC \leftarrow LC \cup \{v\}$ 
(7) Initialize  $U$ ;
(8) endfor
(9) while true
(10)  $\max \leftarrow \infty$ 
(11) foreach  $u$  in  $U$ 
(12)  $\text{delta} \leftarrow \text{CalculateDeltaValue}(u)$ ;
(13)  $t \leftarrow u, \max \leftarrow \text{delta}$ ;
(14) if  $\max > 0$ 
(15)  $LC \leftarrow LC \cup \{t\}$ ;
(16) Update  $N$ ;
(17) else
(18) break;
(19) endif
(20) endfor
(21)  $LCS \leftarrow LCS \cup \{LC\}$ 
(22) endfor
(23) return LCS
End

```

ALGORITHM 3: LCD.

TABLE 1: Information of benchmark networks.

ID	$N$	$k$	$\max k$	$\min c$	$\max c$	$\mu$
S1	1000	10	50	10	50	0.1-0.9
S2	1000	10	50	20	100	0.1-0.9
S3	5000	10	50	10	50	0.1-0.9
S4	5000	10	50	10	50	0.1-0.9

TABLE 2: Information of real networks.

Name	Nodes	Edges	Communities	Reference
Football	115	616	12	[14]
Polbooks	105	441	3	[15]
Karate	34	78	2	[16]
Adjnoun	112	425	2	[17]
Polblogs	1490	19090	2	[18]

LWP and LS make self-inspection and delete the nodes not satisfying certain conditions while making local community extension. LWP takes  $M$  in (2) as the objective function, while LS takes node similarity factor into comprehensive consideration and can adopt any objective function. Both LMD and LCD-MC start local community detection from a node other than the source node. LMD starts from the local degree center node nearest to the source node, while LCD-MC starts from the maximum cliques where the source node is located. Both LMD and LCD-MC are suitable for different objective functions. In our experiment, LS, LMD, and LCD-MC all took  $M$  as the objective function.

We firstly compared the quality of local communities found by these algorithms. As these algorithms, except LCD-MC, have no ability to identify local communities connected by overlapping nodes, in experimental data, we selected LFR benchmarks [21, 22] with nonoverlapped structure and several labeled real networks, whose information is as shown

in Tables 1 and 2. The meaning of the parameters in Table 1 is described as follows:  $N$ , the number of nodes;  $k$ , the average degree;  $\max k$ , the maximum degree;  $\min c$ , the minimum for the community sizes;  $\max c$ , maximum for the community sizes;  $\mu$ , a mixing parameter, the probability of nodes connected with nodes of external community. It should be pointed out that, to find the only local community structure, in the second step of LCD-MC, the maximum clique with the most number of nodes was selected for extension.

To evaluate the quality of the local communities generated by various methods, we adopt F-Measure score (FM) and normalized mutual information (NMI) [23] as the evaluation indexes.

FM is a commonly used measure for community detection algorithms. Assume  $T$  is the set of node pairs  $(i, j)$ , where nodes  $i$  and  $j$  belong to the same classes in the ground truth,

and  $S$  is the set of node pairs that belong to the same communities generated by an algorithm. Then FM is computed from both the precision and the recall synthetically:

$$FM = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (5)$$

where precision and recall are written as (6). Consider that

$$\begin{aligned} \text{precision} &= \frac{|S \cap T|}{|S|}, \\ \text{recall} &= \frac{|S \cap T|}{|T|}. \end{aligned} \quad (6)$$

NMI is another widely used criterion for measuring the performance of community detection algorithms. Formally, the measurement of NMI can be defined as

$$NMI = \frac{-2 \sum_{i=1}^r \sum_{j=1}^k N_{ij} \log(N_{ij}N/N_i N_j)}{\sum_{i=1}^r N_i \log(N_i/n) + \sum_{j=1}^k N_j \log(N_j/N)}, \quad (7)$$

where  $N$  is the confusion matrix,  $N_{ij}$  is the number of nodes both in the  $i$ th class and the  $j$ th cluster,  $r$  and  $k$  are the number of classes and clusters, respectively, and  $N_i$  and  $N_j$  are the number of nodes in the  $i$ th class and the  $j$ th cluster, respectively.

**4.2. Synthetic Networks.** Figures 2, 3, 4, and 5 are the experimental results on the synthetic networks S1, S2, S3, and S4 conducted by the five local community detection algorithms. Each figure includes four subgraphs and the abscissa of each subgraph represents the value of mixed parameter  $\mu$ . The larger the value, the less distinct the community structure; meanwhile it also shows the greater difficulty in finding corresponding community. The ordinates of each subgraph represent the evaluation indexes, which, from left to right and from upper to lower, are precision, recall, FM, and NMI.

From the experimental results, we can acquire the following observation.

- (1) Comparison of LWP and LS with Clauset. It can be seen that, compared with the basic algorithm, both LWP and LS can only focus on either precision or recall. That means that when a high precision can be achieved in local community detection, the recall is often relatively low, and vice versa. Viewing from FM and NMI comprehensive evaluation indexes, neither algorithm can guarantee finding a better local community than that by Clauset. Therefore, it can be known that the critical factor in local community detection is not the self-inspection of the community detected.
- (2) Comparison of LCD-MC and LMD with Clauset. It can be seen that, in terms of the four indexes, both these algorithms can achieve better results than those by Clauset to various extents, indicating that there are really certain restrictions in starting local community detection from the initial node. Viewing from the

TABLE 3: Results on real networks.

Networks	Index	Clauset	LWP	LS	LMD	LCD-MC
Karate	Precision	0.905	0.601	0.852	<b>0.924</b>	0.908
	Recall	0.585	0.278	0.375	0.599	<b>0.658</b>
	FM	0.675	0.329	0.435	0.694	<b>0.744</b>
	NMI	0.31	0.125	0.157	0.343	<b>0.397</b>
Football	Precision	0.665	0.588	0.59	0.723	<b>0.9</b>
	Recall	0.743	0.543	0.554	0.848	<b>0.895</b>
	FM	0.691	0.475	0.483	0.77	<b>0.896</b>
	NMI	0.594	0.301	0.308	0.665	<b>0.841</b>
Polbooks	Precision	0.772	<b>0.824</b>	0.797	0.768	0.787
	Recall	0.477	0.203	0.221	<b>0.784</b>	0.691
	FM	0.52	0.24	0.259	<b>0.756</b>	0.696
	NMI	0.312	0.138	0.14	<b>0.459</b>	0.429
Adjnoun	Precision	0.541	0.409	0.278	0.478	<b>0.524</b>
	Recall	0.24	0.15	0.187	0.615	<b>0.669</b>
	FM	0.224	0.128	0.155	0.431	<b>0.48</b>
	NMI	0.01	<b>0.047</b>	0.044	0.005	0.003
Polblogs	Precision	0.87	0.694	0.616	0.913	<b>0.917</b>
	Recall	0.045	0.205	0.207	0.681	<b>0.722</b>
	FM	0.062	0.211	0.212	0.744	<b>0.788</b>
	NMI	0.058	0.075	0.073	0.431	<b>0.454</b>

experimental results, compared with LMD algorithm which starts the detection from local degree center point, the LCD-MC algorithm that starts from node maximum clique is more effective.

- (3) As a whole, LCD-MC achieves the best results of all four evaluation indexes. On both small community networks (S1 and S3) with a mixed parameter  $\mu$  smaller than 0.6 and large community networks (S2 and S4) with a mixed parameter  $\mu$  smaller than 0.5, LCD-MC could find out the local community structure of each node almost fully correctly. On a highly mixed network, for example, with a  $\mu$  of 0.8 or 0.9, neither LCD-MC nor the other algorithms could obtain ideal results. This just conforms to the real condition that network community structure is not distinct.

To sum up, LCD-MC can find local communities with better quality on synthetic networks compared with the other representative local community detection algorithms.

**4.3. Real Networks.** To further verify the performance of LCD-MC, we compare it with the other algorithms on real networks and show the comparison results in Table 3. The bold digits are the maximum value of local community quality of each algorithm for the related evaluation index. Of them, precision and recall only reflect one aspect of algorithm performance, while FM and NMI take algorithm performance into a comprehensive consideration and, therefore,

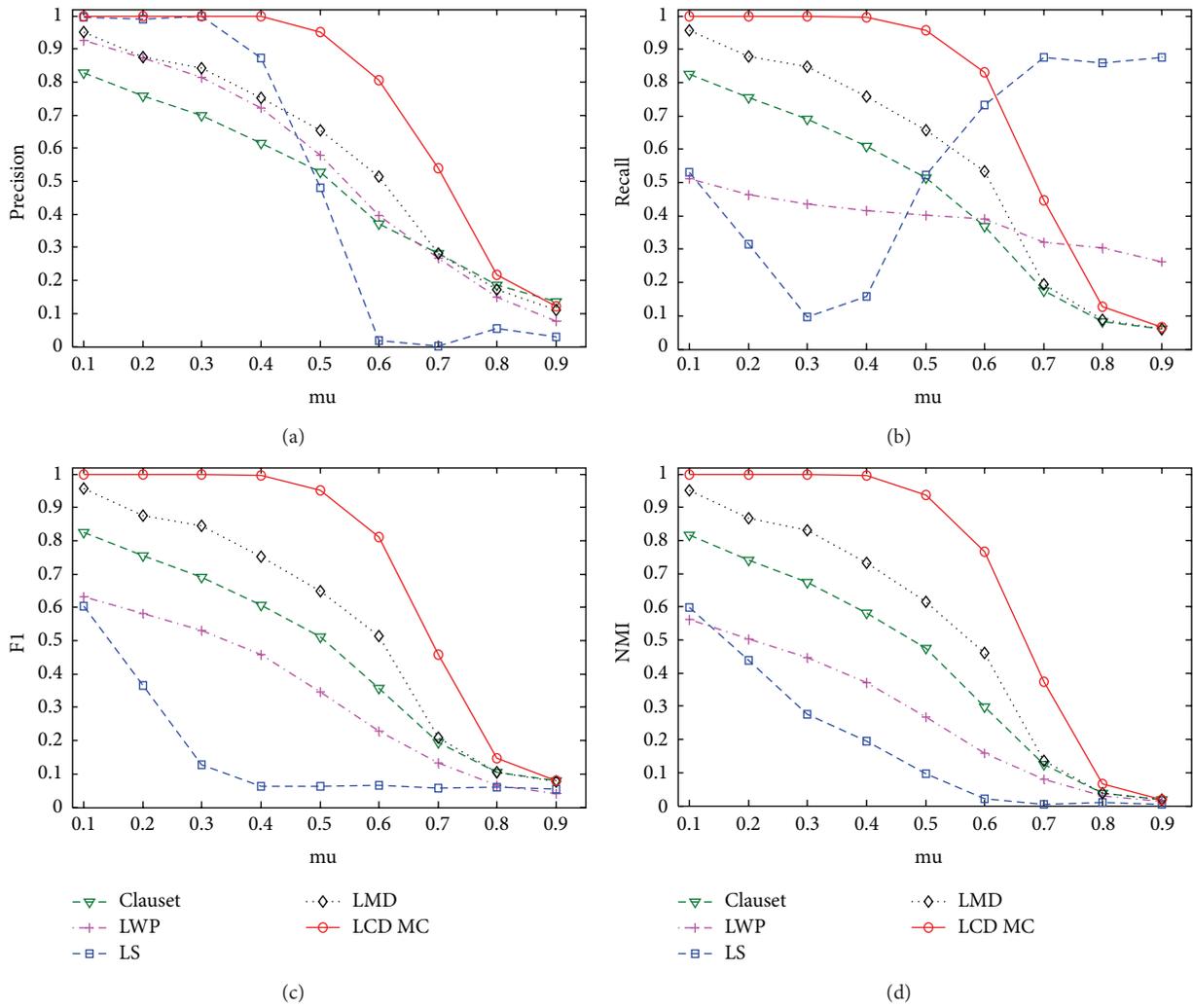


FIGURE 2: Results on SI.

TABLE 4: The real community of the synthetic community.

Community ID	Community elements
1	{1, 10, 15, 20, 21, 28, 35, 38, 46, 49, 51, 59, 61, 63, 66, 69, 73, 74, 80, 82, 87}
2	{5, 8, 22, 24, 25, 26, 29, 31, 32, 34, 37, 39, 42, 45, 47, 55, 56, 57, 58, 62, 70, 84, 86, 89, 96, 97}
3	{0, 2, 3, 4, 6, 9, 11, 13, 14, 30, 41, 44, 48, 50, 53, 54, 56, 64, 68, 76, 81, 83, 90, 91, 94, 95, 98}
4	{2, 7, 12, 16, 17, 18, 19, 23, 27, 28, 33, 36, 37, 40, 43, 52, 60, 65, 67, 71, 72, 75, 77, 78, 79, 85, 88, 92, 93, 99}

TABLE 5: Local community result of Clauset on the synthetic network.

Node ID	Community elements
2	{0, 2, 3, 4, 6, 9, 11, 13, 14, 30, 41, 44, 48, 50, 53, 54, 56, 64, 68, 76, 81, 83, 90, 91, 94, 95, 98}
28	{5, 8, 22, 24, 25, 26, 28, 29, 31, 32, 34, 37, 39, 42, 45, 47, 51, 55, 56, 57, 58, 62, 70, 84, 86, 89}
37	{5, 8, 22, 24, 25, 26, 29, 31, 32, 34, 37, 39, 42, 45, 47, 55, 56, 57, 58, 62, 70, 84, 86, 89, 96, 97}
56	{5, 8, 22, 24, 25, 26, 29, 31, 32, 34, 37, 39, 42, 45, 47, 55, 56, 57, 58, 62, 70, 84, 86, 89, 96, 97}

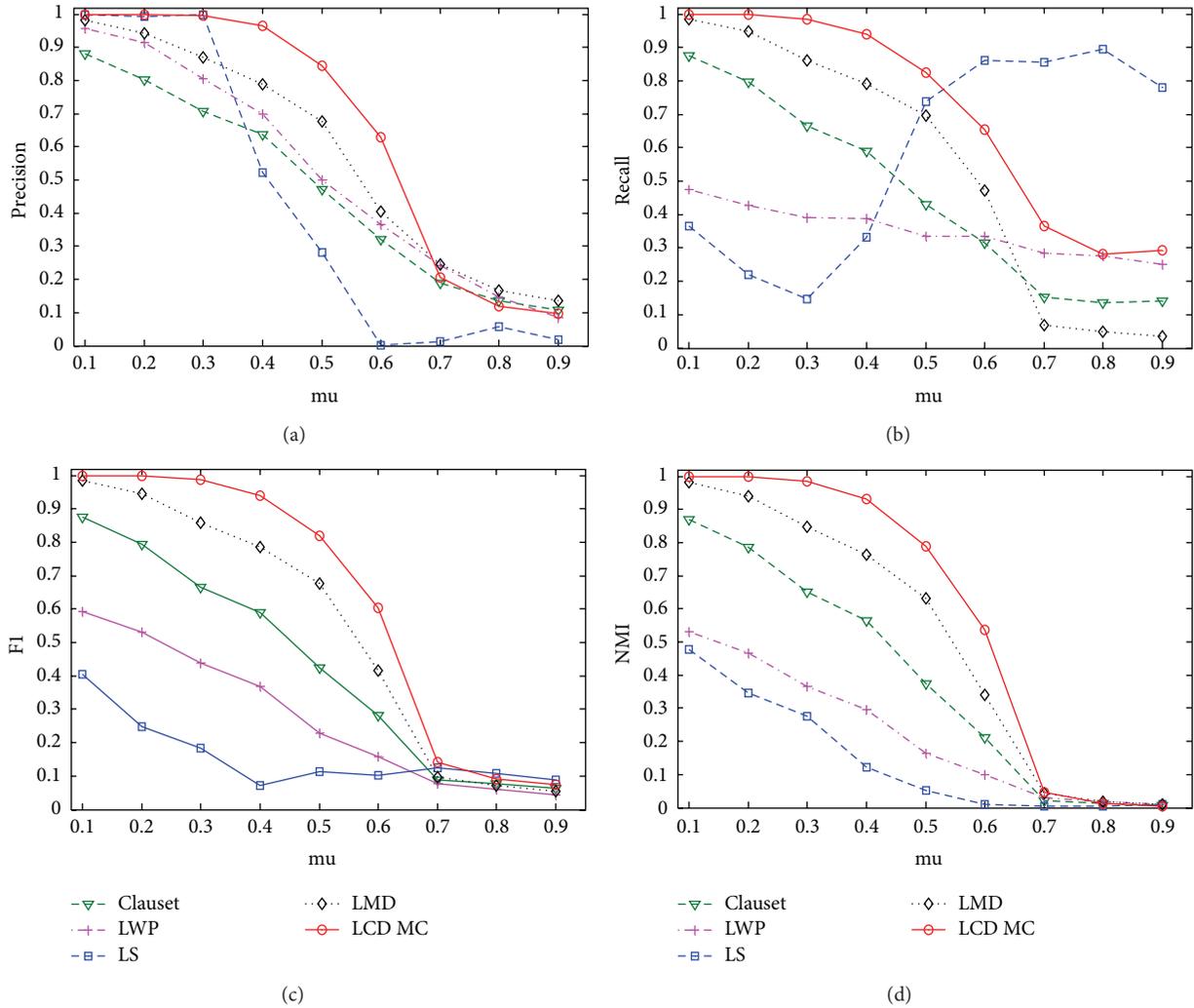


FIGURE 3: Results on S2.

TABLE 6: Local community result of LCD-MC on the synthetic network.

Node ID	Community elements
2	{2, 7, 12, 16, 17, 18, 19, 23, 27, 33, 36, 40, 43, 52, 60, 65, 67, 71, 72, 75, 77, 78, 79, 85, 88, 92, 93, 99} {0, 2, 3, 4, 6, 9, 11, 13, 14, 30, 41, 44, 48, 50, 53, 54, 56, 64, 68, 76, 81, 83, 90, 91, 94, 95, 98}
28	{1, 10, 15, 20, 21, 28, 35, 38, 46, 49, 51, 59, 61, 63, 66, 69, 73, 74, 80, 82, 87} {2, 7, 12, 16, 17, 18, 19, 23, 27, 28, 33, 36, 40, 43, 52, 60, 65, 67, 71, 72, 75, 77, 78, 79, 85, 88, 92, 93, 99}
37	{5, 8, 22, 24, 25, 26, 29, 31, 32, 34, 37, 39, 42, 45, 47, 55, 56, 57, 58, 62, 70, 84, 86, 89, 96, 97} {2, 7, 12, 16, 17, 18, 19, 23, 27, 33, 36, 37, 40, 43, 52, 60, 65, 67, 71, 72, 75, 77, 78, 79, 85, 88, 92, 93, 99}
56	{5, 8, 22, 24, 25, 26, 29, 31, 32, 34, 37, 39, 42, 45, 47, 55, 56, 57, 58, 62, 70, 84, 86, 89, 96, 97} {0, 2, 3, 4, 6, 9, 11, 13, 14, 30, 41, 44, 48, 50, 53, 54, 56, 64, 68, 76, 81, 83, 90, 91, 94, 95, 98}

are of more comparison significance. We can see that LCD-MC achieved the best results on Karate, Football, Polblogs, and Adjnoun networks (all algorithms achieved a very low NMI value on Adjnoun, which can be neglected). Though, on Polbooks network, LCD-MC results are not the best, they are only slightly inferior to the best results achieved by LMD algorithm.

To sum up, LCD-MAC also achieved better experimental results on real networks compared with the other local community detection algorithms.

**4.4. Local Communities of Overlapped Node.** We verified the ability of LCD-MC to identify multiple local communities where overlapped nodes are located on a LFR synthetic

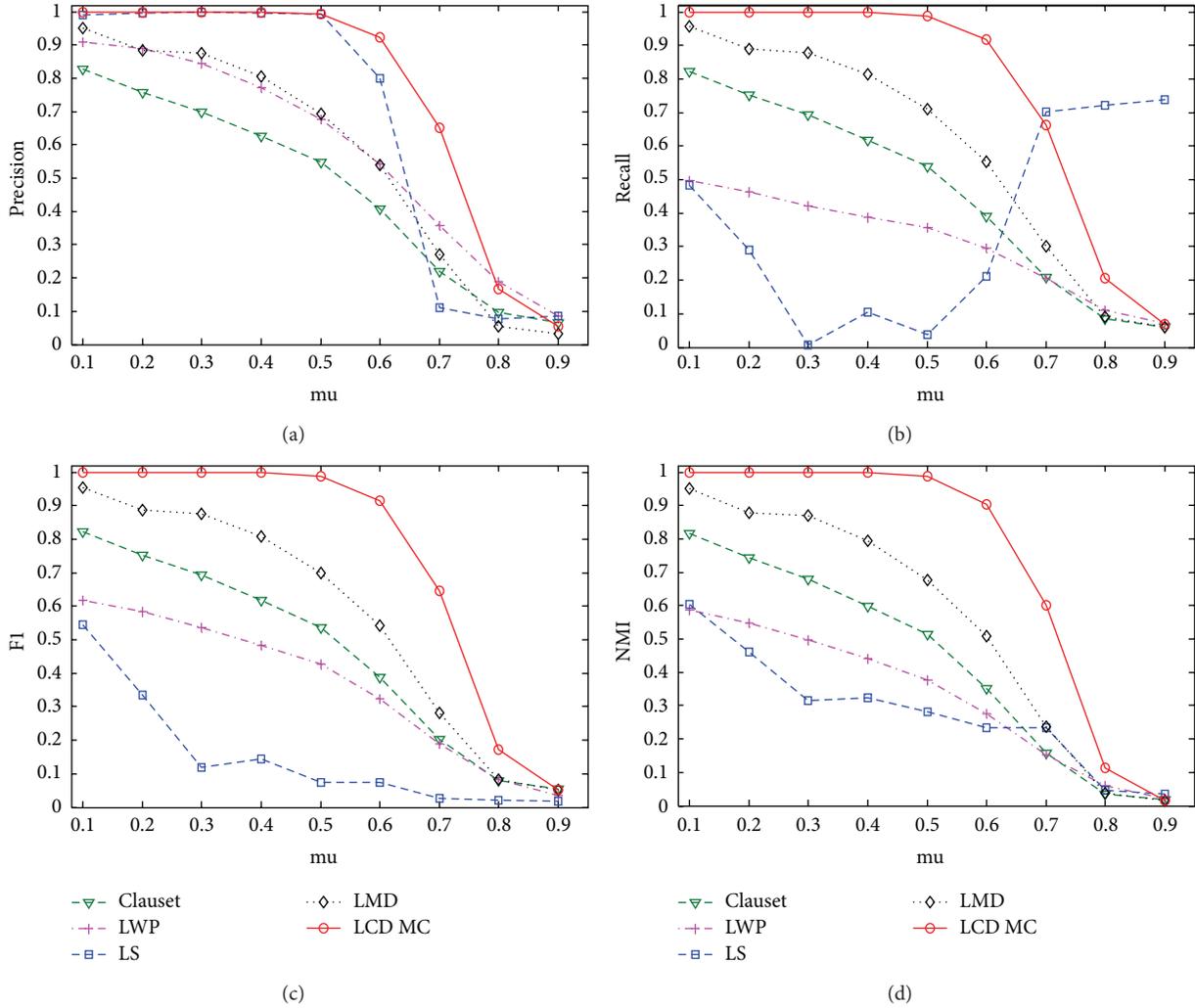


FIGURE 4: Results on S3.

network. The configuration parameters of the LFR synthetic network are  $N = 100$ ,  $k = 10$ ,  $\max k = 20$ ,  $\min c = 20$ ,  $\max c = 30$ ,  $\mu = 0.1$ ,  $on = 4$ , and  $om = 2$ , where  $on$  represents the number of the overlapping nodes, and  $om$  represents the number of memberships of the overlapping nodes. The corresponding network layout is as shown in Figure 6. Table 4 shows the corresponding community distribution, in which 2, 28, 37, and 56 are overlapped nodes and each node connects two communities.

As Clauset, LWP, LS, and LMD algorithms can obtain only one local community from each node; we selected Clauset as their representative in comparison with LCD-MC algorithm. The results of local community detection on the network depicted in Figure 6 by Clauset and LCD-MC are shown in Tables 5 and 6, respectively. It can be seen that Clauset only found out one local community from each overlapped node, because Clauset algorithm can only extend the current only local community according to objective function  $R$ . LCD-MC effectively found out two local communities from each overlapped node. This is because,

in an initialization, LCD-MC uses maximum cliques as candidate local communities instead of only one node. Take node 2 as an example. In initialization, its maximum clique sets included  $\{2, 79, 93\}$ ,  $\{2, 79, 99\}$ , and  $\{2, 83, 94\}$ . Of them, the maximum clique  $\{2, 79, 93\}$  was used as the initial local community for extension and, as a result, community 4 was found, as shown in Table 4. At this stage, the maximum clique  $\{2, 79, 99\}$  was already included in this local community, but the maximum clique  $\{2, 83, 94\}$  was still outside it. Therefore, LCD-MC started from  $\{2, 83, 94\}$  to continue the search for new local community and, then, found two local community structures connected with node 2. Therefore, LCD-MC has the ability to find out multiple local communities connected by overlapped nodes.

## 5. Conclusion

In this paper, we propose a novel local community detection algorithm for large complex networks based on maximum cliques extension (LCD-MC). This algorithm firstly adopts

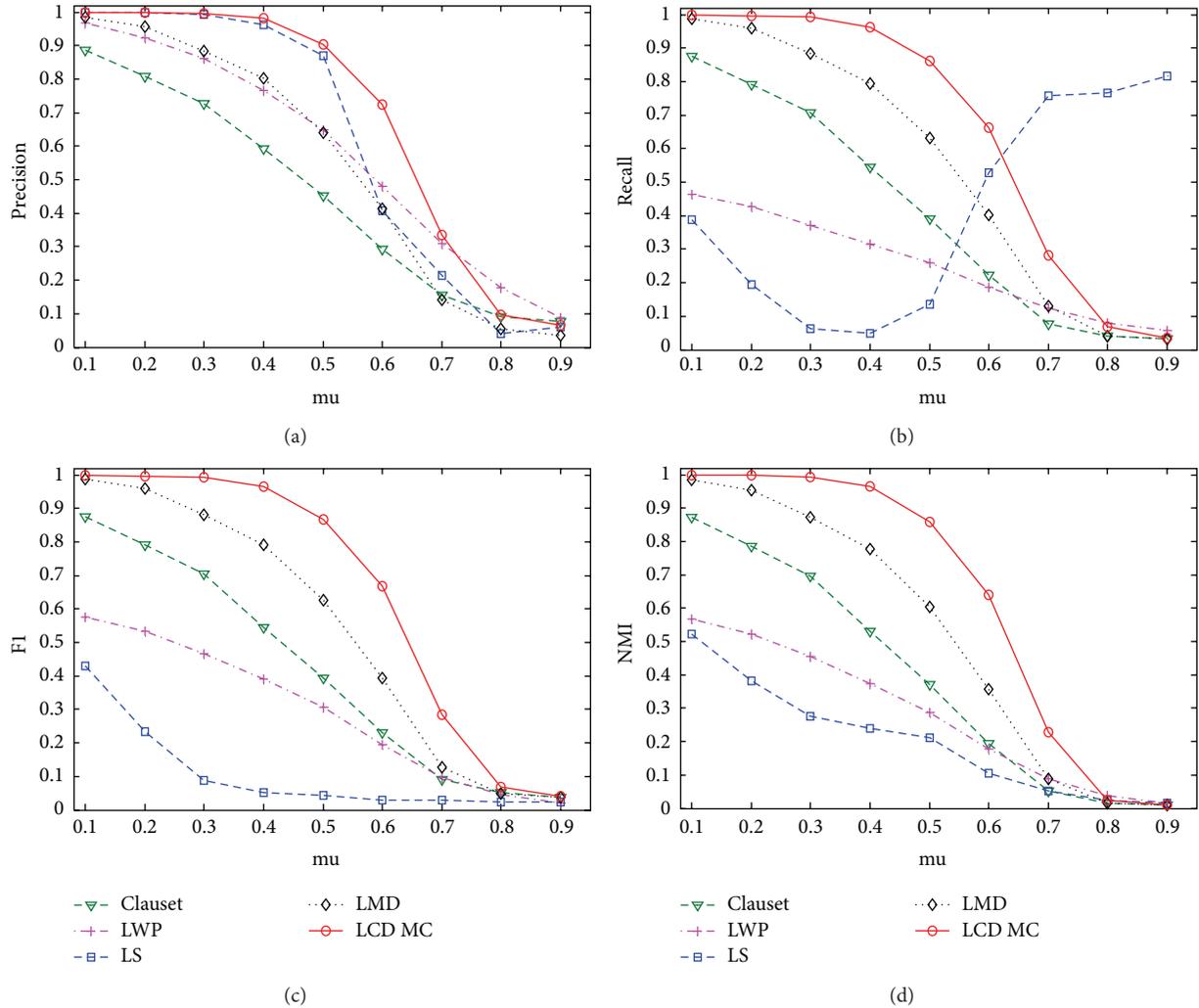


FIGURE 5: Results on S4.

the idea of the Bron-Kerbosch algorithm to find out all maximum cliques containing the source node in the network by recursion, then, takes an arbitrary maximum clique satisfying certain conditions as the initial local community, and, by continuous exploring the neighbor area around the current local community, continuously adds conforming nodes to the local community structure till there is not any conforming node. LCD-MC algorithm is most characterized by starting the extension from the maximum clique of a given node instead of starting directly from the given node. In this way, it avoids the deviation of community extension and can identify multiple local community structures connected by an overlapped node. We compared LCD-MC with some representative algorithms such as Clauset, LWP, LS, and LMD on various synthetic and real networks. The experimental results demonstrate that LCD-MC algorithm can find local communities with better quality on both synthetic and real networks. Moreover, the experimental results on a synthetic network with known overlapped community structures indicate that the LCD-MC algorithm has the ability to identify

multiple local communities where the overlapped node is located.

### Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interest of any nature or kind in any product, service, or company that could be construed as influencing the position presented in, or the review of, the paper.

### Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (Grant no. 2012AA0622022 and Grant no. 2012AA011004), the Doctoral Fund of Ministry of Education of China (Grant no. 20100095110003 and Grant no. 20110095110010), the State

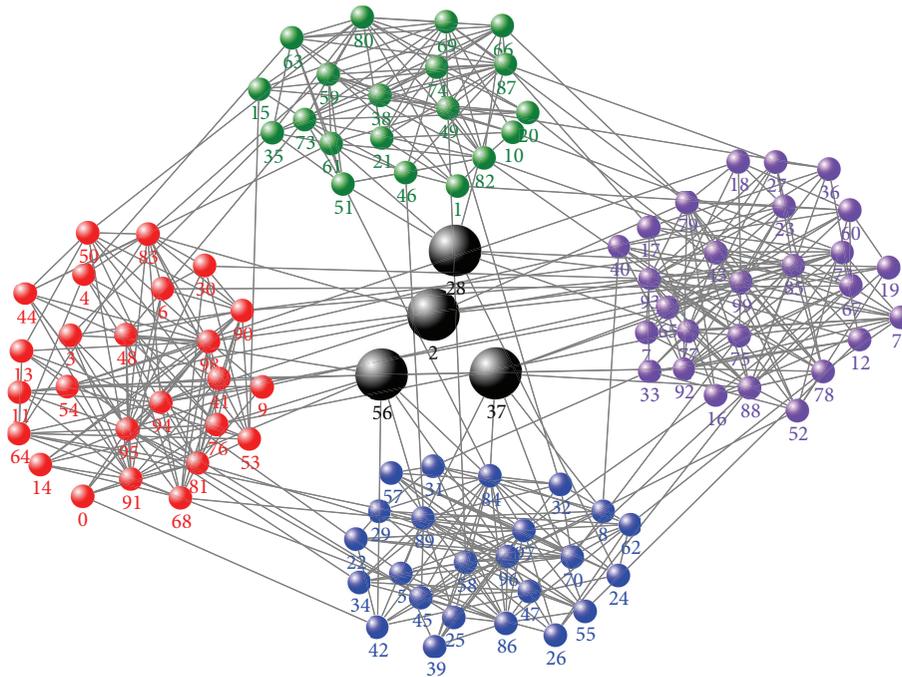


FIGURE 6: An example of synthetic network. Created with NodeXL (<http://nodexl.codeplex.com/>).

863 projects (Grant no. 2012AA011004), and the Graduate Research and Innovation Projects in Jiangsu Province (Grant no. CXZZ12\_0934).

## References

- [1] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Statistical Analysis and Data Mining*, vol. 4, no. 5, pp. 512–546, 2011.
- [2] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.
- [3] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 631–640, April 2010.
- [4] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks the state of the art and comparative Study," *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–37, 2013.
- [5] D. Liu, D. Jin, D. He, J. Yang, and B. Yang, "Community mining in complex networks," *Journal of Computer Research and Development*, vol. 50, no. 10, pp. 2140–2154, 2013.
- [6] M. Coscia, G. Rossetti, and F. Giannotti, "A local-first discovery method for overlapping communities," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pp. 615–623, ACM, August 2012.
- [7] G. Wen, Z. Duan, G. Chen, and X. Geng, "A weighted local-world evolving network model with aging nodes," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 21–22, pp. 4012–4026, 2011.
- [8] X. Geng and G. Wen, "Weighted evolving networks with intrinsic strength," *International Journal of Modern Physics C*, vol. 18, no. 9, pp. 1435–1442, 2007.
- [9] A. Clauset, "Finding local community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 72, no. 2, Article ID 026132, 2005.
- [10] 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.
- [11] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, Article ID 033015, 2009.
- [12] Y. J. Wu, H. Huang, Z. F. Hao et al., "Local community detection using link similarity," *Journal of Computer Science and Technology*, vol. 27, no. 6, pp. 1261–1268, 2012.
- [13] Q. Chen, T. T. Wu, and M. Fang, "Detecting local community structure in complex networks based on local degree central nodes," *Physica A*, vol. 392, no. 3, pp. 529–5537, 2013.
- [14] 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.
- [15] <http://www.orgnet.com/>.
- [16] 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.
- [17] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 3, Article ID 036104, 2006.
- [18] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, ACM, August 2005.

- [19] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.
- [20] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.
- [21] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 78, no. 4, Article ID 046110, 2008.
- [22] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 1, Article ID 016118, 2009.
- [23] M. Zhu, F. Meng, and Y. Zhou, "Semisupervised clustering for networks based on fast affinity propagation," *Mathematical Problems in Engineering*, vol. 2013, Article ID 385265, 13 pages, 2013.



# Hindawi

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

