Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 754029, 8 pages
http://dx.doi.org/10.1155/2015/754029

Research Article

Community Clustering Algorithm in Complex Networks Based on Microcommunity Fusion

Jin Qi,[1] Fei Jiang,[1] Xiaojun Wang,[1] Bin Xu,[1] and Yanfei Sun[2]

[1]*Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210003, China*
[2]*School of Automation, Nanjing University of Posts and Telecommunication, Nanjing 210003, China*

Correspondence should be addressed to Yanfei Sun; sunyanfei@njupt.edu.cn

Received 8 January 2015; Revised 1 April 2015; Accepted 2 April 2015

Academic Editor: Joan Serra-Sagrista

With the further research on physical meaning and digital features of the community structure in complex networks in recent years, the improvement of effectiveness and efficiency of the community mining algorithms in complex networks has become an important subject in this area. This paper puts forward a concept of the microcommunity and gets final mining results of communities through fusing different microcommunities. This paper starts with the basic definition of the network community and applies *Expansion* to the microcommunity clustering which provides prerequisites for the microcommunity fusion. The proposed algorithm is more efficient and *has higher solution quality* compared with other similar algorithms through the analysis of test results based on network data set.

1. Introduction

Network community structure is one of the most common and important topological properties of complex networks whose characteristic is that links between the same communities are dense while links between different communities are sparse. The research on the network community mining algorithm has a very important theoretical meaning for analyzing the topology of complex network, understanding its function, finding its hidden patterns, and predicting its behavior which is widely used in social networks, biological networks, and the World Wide Web. The literature [1–4] summarizes the research background, research significance, research status at home and abroad, and current main problems of the complex network clustering method.

Network community clustering algorithm can be divided into intelligent optimization algorithm and heuristic algorithm or the mixture of the two algorithms. The idea of intelligent optimization algorithm is to abstract the community clustering problem into a mathematical problem of calculating the optimal solution, using intelligent optimization algorithm to calculate the optimal solution which is updated by judging preferential conditions of the objective function. The idea of heuristic algorithm is to calculate the community which each node belongs to according to the rules of the algorithm [5–7].

In recent years, with further research and exploration in the complex network community, efficient community clustering algorithms emerge endlessly. The multiobjective discrete particle swarm optimization (MODPSO) [8], one of intelligent optimization algorithms, calculates the optimal scheme for the community clustering by updating two objective functions: NRA and RC. This algorithm has better nonrandomness and executes efficiently. Moreover, research on heuristic algorithms continues to develop; core node fusion algorithms based on data field [9] and betweenness centrality [10] have also received widespread attention.

Radicchi has given characteristics of the network community structure [11]. Links between nodes in the same communities are dense while links between nodes in different communities are sparse. For a network $G = (V, E)$, V represents set of nodes and edges in networks, k_i represents the degree of node i (the number of nodes connected with node i), A represents an adjacency matrix of the network G,

T represents a community adjacency matrix of G (i.e., $T \subset G$), $k_i^{\text{in}} \sum_{i,j \in T} A_{ij}$, and $k_i^{\text{out}} = \sum_{i \in T, j \in T} A_{ij}$.

The definition of strong community is

$$\forall i \in T, \quad k_i^{\text{in}} > k_i^{\text{out}}. \tag{1}$$

The definition of weak community is

$$\sum_{i \in T} k_i^{\text{in}} > \sum_{i \in T} k_i^{\text{out}}. \tag{2}$$

According to this characteristic of the community, we can divide a real community which a node belongs to and split edges which connected with node i into two parts: edges connected with community T and edges disconnected with community T, and the number of edges is k_i^{in} and k_i^{out}, respectively. We can determine the community which nodes belong to by comparing the two values. If $k_i^{\text{in}} > k_i^{\text{out}}$, we can determine which community that node i belongs to. Further analysis shows that if $k_i^{\text{in}} \geq k_i/2$, we can determine that node i belongs to this community.

Newman and Girvan put forward the concept of modularity to measure the quality of network community clustering in paper [12]. Many community clustering algorithms have accepted this concept as an index to measure the quality of community clustering. The formula of modularity Q is given as follows:

$$Q = \frac{1}{2} \sum_{i,j} \left[\left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i, j) \right], \tag{3}$$

where m represents the number of edges in the network, A_{ij} is the adjacency matrix of the network, k_i is the degree of node i, and $\delta(i, j) = 1$ represents that node i and node j belong to the same community while $\delta(i, j) = 0$ represents that node i and node j are not in the same community.

In multiobjective particle swarm algorithm, objective function in single objective particle swarm modularity Q is further replaced with modularity density D and we explore by updating the value of RA (*Ratio Association*) and the RC (*Ratio Cut*). The formula of RA, RC, and D is shown as follows:

$$\text{RA} = \sum_{i=1}^{k} \frac{L(V_i, V_i)}{|V_i|}, \tag{4}$$

$$\text{RC} = \sum_{i=1}^{k} \frac{L(V_i, \overline{V_i})}{|V_i|}, \tag{5}$$

$$D = \text{RA} - \text{RC}, \tag{6}$$

where n denotes the number of nodes in the network, K represents the number of divided communities in the network, V_i is the ith community among divided communities, $|V_i|$ is the number of nodes in the ith community, $\overline{V_i}$ represents the set of nodes which are not in the community i, $L(V_i, V_j) = \sum_{a \in V_i, b \in V_j} A_{ab}$, A is the adjacency matrix of the network, and RA and RC are closely connected with the two measurement indexes (*Conductance* and *Expansion*) of network community clustering mentioned in the paper [13]. *Conductance* denotes the ratio of the number of nodes pointing outside the community to the number of edges of the community. *Expansion* represents the number of edges each node has which point outside the community.

The formula of *Conductance* is

$$f(s) = \frac{c_s}{2m_s + c_s}. \tag{7}$$

The formula of *Expansion* is

$$f(s) = \frac{c_s}{n_s}. \tag{8}$$

In the above formulas, c_s represents the number of links on the boundary of s, m_s denotes the number of links within the community s, and n_s is the number of nodes in community s.

The algorithm of this paper adopts the divide-and-conquer strategy [14]. The nodes in the network are divided into microcommunities with a single node as the core. We can get the community structure through fusing the microcommunities randomly [15]. After finishing the core steps of the algorithm, the final result of the community clustering can be screened out via the index of the modularity density. The clustering of microcommunities and the whole process of the algorithm will be described in Section 2 in detail. Section 3 will make simulation analysis on experimental results of the algorithm.

2. Microcommunity Fusion

The algorithm in this paper constructs microcommunities according to the index *Expansion* during the process of community clustering. In the procedure of microcommunity fusion, it merges and fuses microcommunities according to the definition of strong community.

2.1. Microcommunity. The algorithm is different from other heuristic algorithms. The algorithm divides communities into several basic "microcommunities" in the network and then mergers and fuses these "microcommunities" to get the final results of the network community clustering.

Firstly, N nodes with larger values of degree in the network are selected corresponding to center nodes of N communities in the network. The selected minimum value of degree is called the threshold. By testing different choices of the threshold during the process of the algorithm, we can find that the threshold is larger than the average value of degree of network nodes and the experimental result is ideal when the number of center nodes accounts for about 20% (see Section 3.4) of the total number of nodes. The formula of the choice of threshold Deg is

$$\text{Deg} = N \left[\lfloor n(1 - \lambda) \rfloor \right], \tag{9}$$

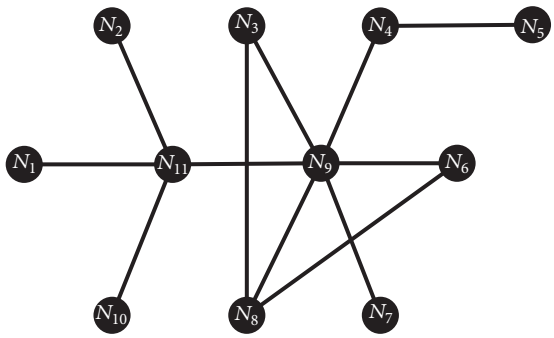

Figure 1: Example of calculation process of *Expansion*.

where $N = \{N_1, N_2, \ldots, N_n\}$ is an array which is in ascending order according to the value of node degree in the network, n is the number of elements in the array, that is, the number of nodes in the network, and the value of λ is 20% in this algorithm.

According to the chosen center node, eligible node in its neighbors is selected to join the microcommunity. This algorithm uses the index *Expansion* summarized in [10] for choice. Because *Expansion* is a nonlinear function, when the number of nodes is small, the change range of the function value cannot meet the expectation. Thus, this paper adjusts the computing method in the index *Expansion* and removes the center nodes and connected edges. The calculation example of *Expansion* is given in Figure 1.

The calculation process of *Expansion* which used N_9 as its center node is given as follows. At the stage of initialization, we set all neighbor nodes of N_9 as a microcommunity which sets N_9 as its center node and the value of *Expansion* is 4/6. The algorithm traverses each neighbor node and calculates the value of *Expansion* after removing it out of the microcommunity. If the value becomes small, the node will be removed. Otherwise, it calculates the next neighbor node. In the example, the change process of *Expansion* value of each node in the network diagram is given in Table 1. One of the initial nodes traversed is randomly selected as N_3. EXP is the value of *Expansion* before removing the node. EXP_NEXT is the value of *Expansion* after removing the node.

According to the information from the table, nodes which set N_9 as center node of the microcommunity are N_3, N_6, N_7, N_8, and N_9. Compared with standard *Expansion*, the *Expansion* used to screen out the node from the microcommunity is stricter in computational condition. It is conducive to make the structure of microcommunity stable and the change of nodes is more explicit during the process of the microcommunity fusion.

2.2. Algorithm Flow. In this algorithm, each center node clustered is used as the core node of a microcommunity. The algorithm merges microcommunities by comparing the close level of links between different core nodes. a and b are core nodes, k_a represents the degree of node a, k_b represents the degree of node b, x_a represents the assigned community number of node a, x_b represents the assigned community number of node b, A is the adjacency matrix of two nodes, a and b have some overlapping neighbor nodes, and n is the number of those overlapping neighbor nodes. If $A_{ab} = 0$, we do not take measures to deal with both nodes. If $A_{ab} = 1$, n will be calculated.

Table 1: The selection of nodes in the microcommunity.

NODE	EXP	EXP_NEXT	EXP_NEXT < EXP?	REMOVING
N_3	4/6	1	NO	
N_4	4/6	3/5	YES	Remove
N_6	3/5	1	NO	
N_7	3/5	3/4	NO	
N_8	3/5	5/4	NO	
N_{11}	3/5	0	YES	Remove

If

$$n \geq \frac{k_a}{2}, \tag{10}$$

then $x_a = x_b$. All nodes of the microcommunity are updated synchronously.

If

$$n \geq \frac{k_b}{2}, \tag{11}$$

then $x_b = x_a$. All nodes of the microcommunity are updated synchronously.

After completing the fundamental fusion of microcommunities, the node without clustering will be classified. The proportion of the number of each neighbor node of the undetected nodes in community numbers is checked. Then, the node will be added to the community which has the largest proportion. The operation of merging is implemented according to the sequence of nodes during the process of searching network nodes three times. But as we know, the relationship of nodes in complex networks is extremely cumbersome. Each node may repeat with more than one node's neighbor nodes. And the ratio of repetition is more than 1/2. Therefore, if we take the ordinal search classification algorithm, some unpredictable extreme situations will emerge. In view of the consideration of the detail, in this paper, the order of search is generated randomly. In the last step of the algorithm, the result which fits the community clustering rules better can be screened out.

The following are the specific steps of the algorithm.

Step 1. Detect nodes in the network to N microcommunities and set the numbers of center nodes as the microcommunity numbers according to formula (8) and (9).

Step 2. Fuse N microcommunities on the basis of the order of the random sequence according to formula (10) and (11).

Step 3. The community which is not detected is added to its most closely linked community.

Step 4. Classify nodes of communities which have not been detected.

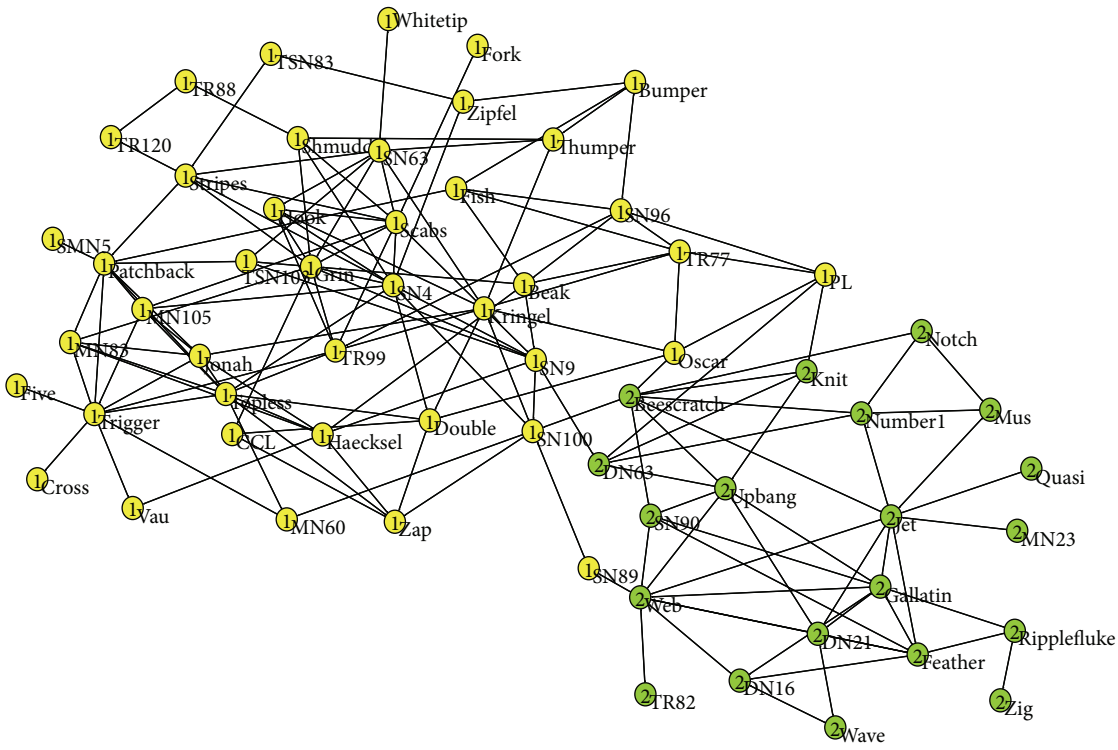

Figure 2: Dolphin Network based on Microcommunity Fusion algorithm.

Step 5. Save the result of classification, compute the module density D, and save it after detecting community.

Because the different order of merging network nodes can lead to different results, the clustering result with larger value of module density D is used as the final result of the community clustering according to formula (6).

During the process of conducting the core steps, Steps 2 and 3, of the algorithm, both steps search the network nodes once, respectively. The time complexity of the algorithm is $O(n^2)$.

3. Simulations and Analysis

The algorithm of this paper is written in *JAVA*. The hardware environment of running the program is *Inter (R) Core (TM) i5-4200U CPU*, *1.60 GHZ*, and *4 GB RAM*. The software environment is *Microsoft Windows 8.1 operating system*, *jdk* 1.7, and *Eclipse software development environment*.

In order to analyze the quality of network community clustering easily, this paper adopts the so-called *Normalized Mutual Information* (NMI) index described in [16] to compare the actual clustering result with the clustering result of this algorithm. NMI is commonly used to estimate the similarity between the true clustering results and the detected ones. Two vectors, A and B, are inputted during the process of comparison. The ith bit of the vector represents the class of the ith node. The NMI (A, B) is then defined as follows:

$$\mathrm{NMI} = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{i=1}^{C_A} C_{i.}\log\left(C_{i.}/N\right) + \sum_{j=1}^{C_B} C_{.j}\log\left(C_{.j}/N\right)}, \quad (12)$$

where C_A (C_B) is the number of clusters in vector A (B), C is the mixing matrix which consists of vector A and vector B, C_{ij} is the number of elements shared in common by the gth classification of vector A and by the jth classification of vector B, $C_{i.}$ ($C_{.j}$) is the sum of elements of C in *row* i (column j), and N is the number of nodes of the network. The value of NMI (A, B) is in the interval $[0, 1]$. If NMI $(A, B) = 1$, then $A = B$. If NMI $(A, B) = 0$, then A and B are totally different.

This paper conducts the test on Dolphin Networks, Football Networks, Karate Networks, and so on. The clustering result of the algorithm in this paper is better than other algorithms by analyzing the experimental results. At the same time, this algorithm has higher execution efficiency.

3.1. Experimental Data Analysis of Dolphin Networks. Data set profile of Dolphin Networks [17] is shown in Table 2 and Figure 2.

Each node represents a bottlenose dolphin in the data set. By observing the living habits of these dolphins for a long time, their study found that these dolphins show a specific pattern of contact and construct a social network containing 62 nodes. If two dolphins do something together frequently, there will be an edge between the two corresponding nodes in the network.

The algorithm of this paper conducts the community clustering on Dolphin Networks and sets the maximum value of the module density: $D = 4.326$. This clustering result is as follows: the value of the module degree Q is 0.374 and the value of NMI is 1.0. This result is the same as the actual community clustering.

Table 2: Dolphin Networks data set properties.

Properties	Values
Number of nodes	62
Average clustering coefficient	0.303
Number of edges	159
Diameter	8
Number of triangles	95
Average shortest path length	3.357

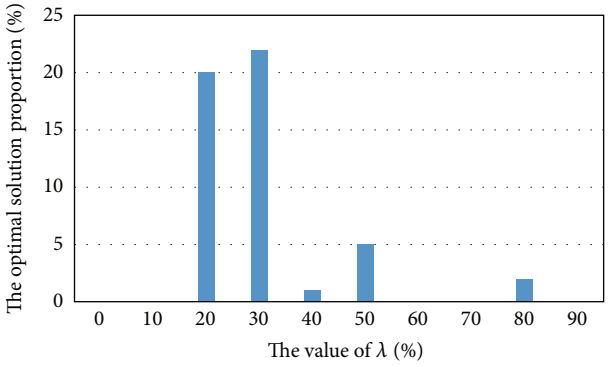

Figure 3: Proportion of real clustering result calculated from different λ.

As already stated in Section 2.1, the threshold selected in this data set is $N[62 * (1–20\%)] = 7$; that is, the node whose degree is equal or greater than 7 is chosen as the core node. During the investigation of the data set, we chose multiple parameters for test. Figure 3 gives the comparison of real clustering results from 10 groups of parameter calculation in which λ choose the value from 0 to 90%. As shown in the diagram, when $\lambda = 20\%$ and $\lambda = 30\%$, the real clustering result occupies the largest proportion. From the calculation, we find that when $\lambda = 30\%$, obtained threshold $N[62 * (1–30\%)] = 7$ is the same as the former.

3.2. Experimental Data Analysis of Football Networks. Data set profile of [18] Football Networks is shown in Table 3.

In the network, each node represents a university team which participates in the USA football season in 2000. The edge which links two nodes represents that the corresponding two teams once had a game at least rather than the relationship between the two teams.

The actual community structure of Football Networks is given in Figure 4. We can get the community clustering result shown in Figure 5 by using the algorithm in this paper. The module degree Q of the actual community clustering of Football Networks is −0.0239 and the module density of the actual community clustering of Football Networks is −100.83. Obviously, the actual networks clustering of Football Networks does not fully comply with the rules of network community clustering. In Figure 4, we can find that all nodes of community 6 cannot meet the basic rules of community clustering. Nodes of community 6 have no connection with each other. But the connection between nodes of community 6 and nodes of other communities is dense. The condition

Table 3: Football Networks data set properties.

Properties	Values
Number of nodes	115
Average clustering coefficient	0.403
Number of edges	613
Diameter	4
Number of triangles	810
Average shortest path length	2.508

Table 4: Karate Networks data set properties.

Properties	Values
Number of nodes	34
Average clustering coefficient	0.588
Number of edges	78
Diameter	5
Number of triangles	45
Average shortest path length	2.408

that a few nodes have less connection with their own community also exists in other communities. It is inevitable for those communities which have a lot of nodes.

Figure 6 also gives the comparison of experimental results when λ choose different parameters. Because of the irrationality of the real clustering in this data set, the diagram only shows the proportion of the modularity density when the value is larger than 1 in experimental results. From the diagram, we can find that when λ choose the value between 20% and 50%, the proportion is large, and the final threshold of the degree is same, so we choose 20% in the experiment.

3.3. Experimental Data Analysis of Karate Networks. This network is a classical data set in the field of social network analysis [19]. In the early 1970s, Zachary, a sociologist, spent two years to observe the social relation network among 34 members of a karate club in an American university. The network consists of 34 nodes. An edge between two nodes indicates that the corresponding members are friends and they contact each other frequently. The network attribute profile is shown in Table 4.

According to this algorithm, we conduct the community clustering to Karate Networks. We set the maximum value of the module density $D = 2.826$. At the same time, NMI = 1.0. This result is the same as the result of the actual community clustering. The topology of Karate Networks community clustering is shown in Figure 7.

Figure 8 gives the contrast diagram of experimental results when λ in this network data set choose different parameters. When $\lambda = 20\%$, the threshold of degree is 6; the real clustering result occupies the largest proportion.

3.4. Summary. Through comparing experimental results using different values of λ in different networks comprehensively, we choose $\lambda = 20\%$ as the final parameter which has good experimental results for most of the networks. In fact, the selection of λ covers a wide range because a threshold

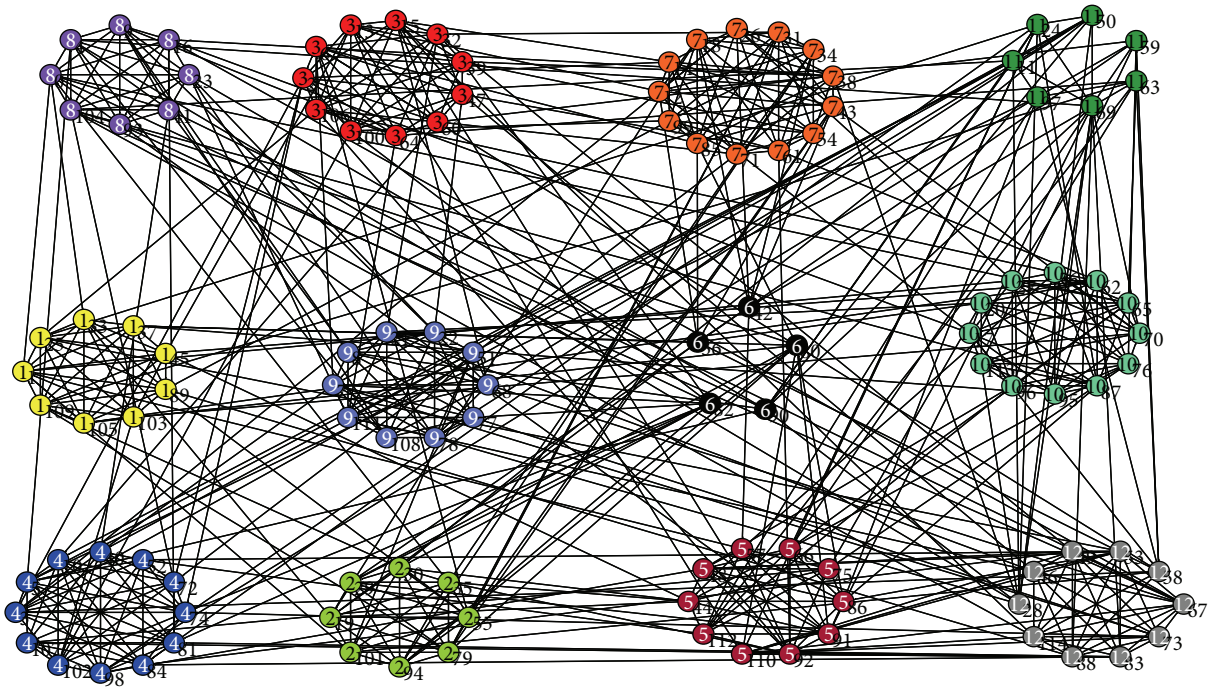

Figure 4: The actual community clustering of Football Networks.

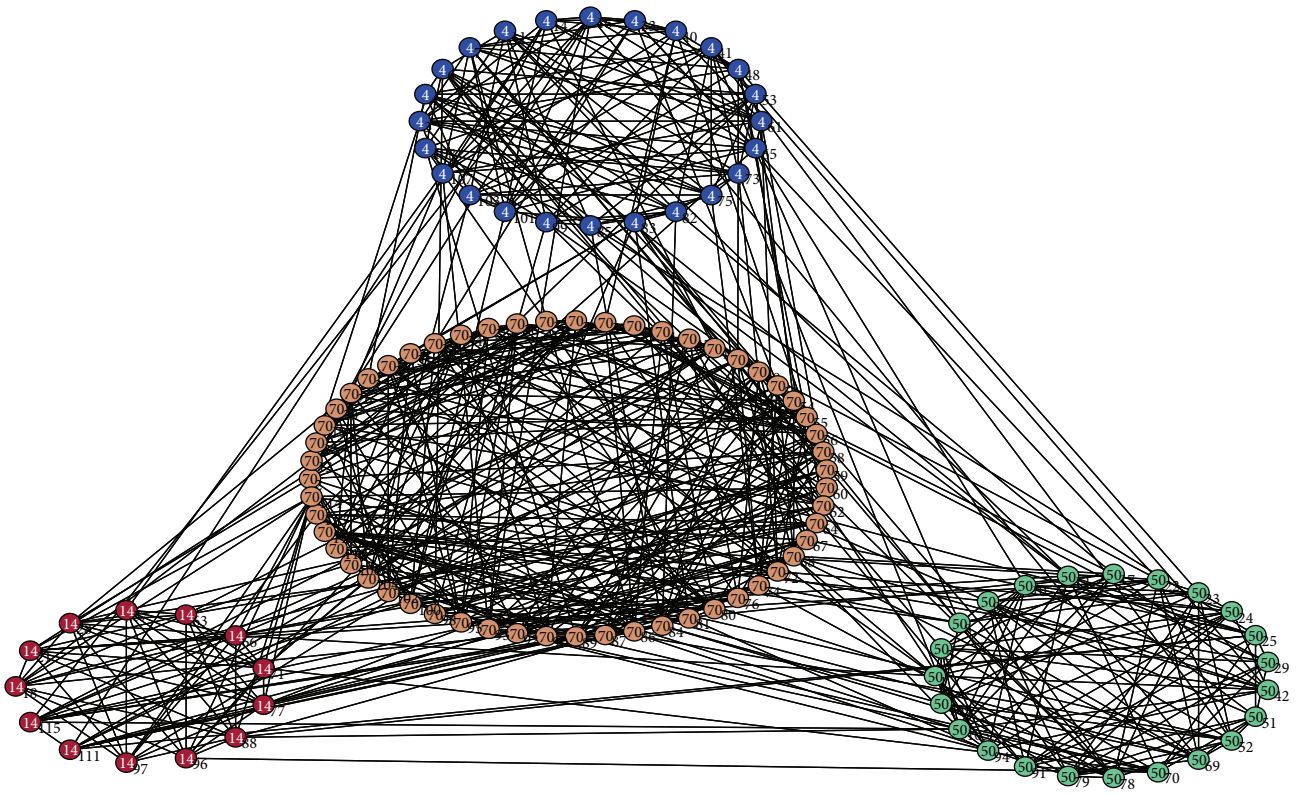

Figure 5: Football Network based on Microcommunity Fusion algorithm.

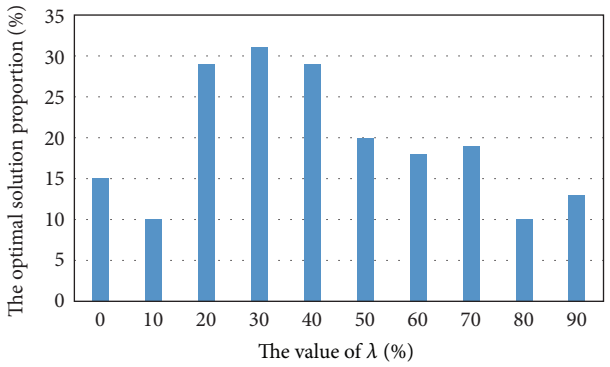

Figure 6: Experimental comparison of each of the λ in data set of Football Netwok data set.

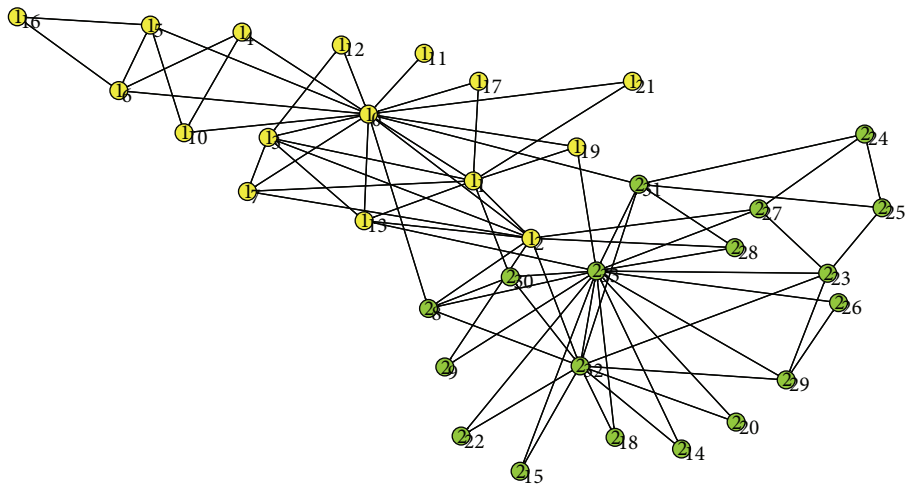

Figure 7: Karate Network based on Microcommunity Fusion algorithm.

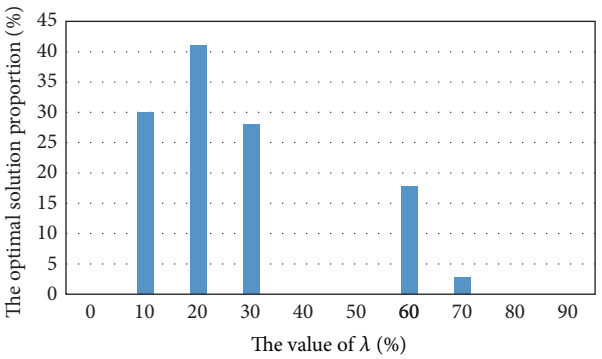

Figure 8: Experimental comparison of each of the λ in data set of Karate Network data set.

may correspond to multiple λ, while $\lambda = 20\%$ is covered in parameters in most of the better experimental results.

4. Conclusions

There are many kinds of algorithms for community clustering in complex networks. All of them, however, not only have advantages but also have drawbacks. According to the degree of nodes in the network and *Expansion*, the algorithm proposed in this paper clusters several microcommunities and gets the final community structure of the network by merging microcommunities. The clustering can be implemented by generating 100 random sequences when merging core nodes, and then the result with the maximum value of the modularity density will be selected as the final result of clustering. Experiments show that sieve method used in this paper can efficiently find the result which is very close to the result of community structure in real networks. According to the basic principle of the network community clustering, the algorithm can give the better community structure in an efficient way.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper was supported by Hi-tech Research and Development of China, 863 Funds (2009AA01Z212), National

Natural Science Foundation of China (61003237 and 61401225), Jiangsu Provincial National Science Foundation (BK20140894) NUPTSF (Grants nos. NY213047 and NY213050), and Higher Education Revitalization Plan Foundation of Anhui (2013SQRL102ZD).

References

[1] S. Kang and D. A. Bader, "Large scale complex network analysis using the hybrid combination of a mapreduce cluster and a highly multithreaded system," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW '10)*, pp. 1–8, Atlanta, Ga, USA, April 2010.

[2] T. Pei, H. Zhang, Z. Li, and Y. Choi, "Survey of community structure segmentation in complex networks," *Journal of Software*, vol. 9, no. 1, pp. 89–93, 2014.

[3] D. W. Zhang, F. D. Xie, D. P. Wang, Y. Zhang, and Y. Sun, "Cluster analysis based on bipartite network," *Mathematical Problems in Engineering*, vol. 2014, Article ID 676427, 9 pages, 2014.

[4] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "Quantitative function for community detection," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 77, no. 3, Article ID 036109, 2008.

[5] B. Yang, D.-Y. Liu, J. Liu, D. Jin, and H.-B. Ma, "Complex network clustering algorithms," *Ruan Jian Xue Bao/Journal of Software*, vol. 20, no. 1, pp. 54–66, 2009.

[6] X. Liu, D. Li, S. Wang, and Z. Tao, "Effective algorithm for detecting community structure in complex networks based on GA and clustering," in *Computational Science—ICCS 2007: 7th International Conference, Beijing, China, May 27–30, 2007, Proceedings, Part II*, vol. 4488 of *Lecture Notes in Computer Science*, pp. 657–664, Springer, Berlin, Germany, 2007.

[7] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 389, no. 7, pp. 1493–1500, 2010.

[8] M. G. Gong, Q. Cai, X. W. Chen, and L. J. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82–97, 2014.

[9] Y. H. Liu, J. Z. Jin, Y. Zhang, and C. Xu, "A new clustering algorithm based on data field in complex networks," *Journal of Supercomputing*, vol. 67, no. 3, pp. 723–737, 2014.

[10] C. Tong, J. W. Niu, B. Dai, and Z. Y. Xie, "A novel complex networks clustering algorithm based on the core influence of nodes," *The Scientific World Journal*, vol. 2014, Article ID 801854, 7 pages, 2014.

[11] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[12] 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, Article ID 026113, 15 pages, 2004.

[13] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*, pp. 631–640, ACM, Raleigh, NC, USA, April 2010.

[14] M. Khalilian, F. Z. Boroujeni, N. Mustapha, and M. N. Sulaiman, "K-means divide and conquer clustering," in *Proceedings of the International Conference on Computer and Automation Engineering (ICCAE '09)*, pp. 306–309, IEEE Computer Society, Bangkok, Thailand, May 2009.

[15] R. Green, I. Staffell, and N. Vasilakos, "Divide and Conquer? k-means clustering of demand data allows rapid and accurate simulations of the British electricity system," *IEEE Transactions on Engineering Management*, vol. 61, no. 2, pp. 251–260, 2014.

[16] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *European Physical Journal B*, vol. 38, no. 2, pp. 331–338, 2004.

[17] http://networkdata.ics.uci.edu/data/dolphins/dolphins.zip.

[18] http://networkdata.ics.uci.edu/data/football/football.zip.

[19] http://networkdata.ics.uci.edu/data/karate/karate.zip.