Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 2942054, 12 pages
https://doi.org/10.1155/2018/2942054

Research Article

An Autonomous Divisive Algorithm for Community Detection Based on Weak Link and Link-Break Strategy

Xiaoyu Ding, Jianpei Zhang, Jing Yang, and Yiran Shen

College of Computer Science and Technology, Harbin Engineering University, Heilongjiang 150001, China

Correspondence should be addressed to Jianpei Zhang; zhangjianpei@hrbeu.edu.cn

Received 4 June 2017; Accepted 14 December 2017; Published 15 January 2018

Academic Editor: Sebastian Anita

Divisive algorithms are widely used for community detection. A common strategy of divisive algorithms is to remove the external links which connect different communities so that communities get disconnected from each other. Divisive algorithms have been investigated for several decades but some challenges remain unsolved: (1) how to efficiently identify external links, (2) how to efficiently remove external links, and (3) how to end a divisive algorithm with no help of predefined parameters or community definitions. To overcome these challenges, we introduced a concept of the weak link and autonomous division. The implementation of the proposed divisive algorithm adopts a new link-break strategy similar to a tug-of-war contest, where communities act as contestants and weak links act as breakable ropes. Empirical evaluations on artificial and real-world networks show that the proposed algorithm achieves a better accuracy-efficiency trade-off than some of the latest divisive algorithms.

1. Introduction

The study of networks is now one of the most active interdisciplinary research fields [1, 2]. In the research of computer science and sociology, complex systems are abstracted as networks or graphs. The basic components of the network are nodes and links. Nodes represent entities of interest. Links represent associations among entities. Community structure is one of the most important properties of complex systems, and community detection is an effective approach to study this property. The goal of detecting community structure is to get an appropriate classification where the links to the nodes with the community are dense, while the links to the nodes out of the community are sparse [3–7].

Nowadays, different community detection algorithms have been proposed [1, 2], such as divisive algorithms [8–11], clustering algorithms [5, 12–15], modularity optimization algorithms [16–20], and label propagation algorithms [21–24]. This paper focuses on the study of divisive algorithms which separate communities by detecting and removing links. Girvan and Newman [25] proposed a significant algorithm based on the betweenness which can identify external links [10]. However, as a global centrality index, the calculation of betweenness is time-consuming and each iteration of the algorithm removes only one link from the network. To improve the efficiency of divisive algorithms, Radicchi et al. [9] proposed the edge-clustering coefficient which is a local centrality index. Based on the edge-clustering coefficient, the proposed algorithm can remove multiple links from the network at each iteration. However, the result of the algorithm is a mass of trivial partitions. To get a trade-off between accuracy and efficiency, Yang et al. [11] proposed an algorithm based on closed walks. However, the termination of the algorithm depends on the quality function modularity [16, 26, 27].

This paper focuses on three challenges: (1) how to detect external links efficiently, (2) how to remove external links efficiently, and (3) how to end a divisive algorithm with no help of predefined parameters or community definitions. Actually, if communities can distinguish between internal and external links, then communities can remove external links, keep internal links, and define themselves. Based on this idea, we present a concept of the weak link and autonomous division. The implementation of the autonomous divisive (AD) algorithm adopts a new link-break strategy similar to a tug-of-war contest.

We summarize the main contributions of this paper as follows:

(i) We propose a concept of the weak link. We define the weak link as a link which locates on the boundary of a community and is likely to connect another community. By removing weak links, communities get disconnected from each other. The experimental results on both artificial and real-world networks show that the weak link improves the efficiency for detecting external links.

(ii) We propose a link-break strategy based on the weak link. The link-break strategy achieves a great efficiency by detecting and removing multiple weak links at each iteration of the proposed algorithm. Based on the link-break strategy, the number of iterations of a divisive algorithm can be reduced.

(iii) We propose an autonomous divisive algorithm based on the weak link and link-break strategy. "Autonomous" means the proposed divisive algorithm does not require parameters, nontopological information, and community definition. The proposed algorithm can end with no help of predefined parameters or community definitions.

The rest of the paper is organized as follows. Section 2 reviews related works of divisive algorithms. Section 3 introduces the proposed definitions and algorithm. We test our algorithm and compare it with other divisive algorithms in Section 4. Section 5 concludes our study.

2. Related Works

2.1. Betweenness (GN) Algorithm. Girvan and Newman [25] proposed the GN algorithm. In their work, they proposed betweenness focusing on the links that are most "between" communities. Each iteration of GN removes the link with the highest betweenness and then recalculates the betweenness of all the links affected by the removal. For further study, they considered alternative definitions of betweenness. Experimental results showed that the proposed algorithm based on the shortest path betweenness shows the best performance [10].

2.2. Distance Dissimilarity (DD) Algorithm. Zhou [8] proposed the DD algorithm to quantify the differences between communities. Zhou introduced the dissimilarity index to measure the possibility that two adjacent nodes belong to the same community. Besides, Zhou also introduced a resolution threshold value known as the dissimilarity threshold. At each iteration of DD, the value of dissimilarity threshold decreases differentially. Based on the dissimilarity threshold, DD can remove multiple links at each iteration and get hierarchically organized communities characterized by upper and lower dissimilarity thresholds [8].

2.3. Information Centrality (IC) Algorithm. Fortunato et al. [28] proposed the IC algorithm based on information centrality defined as the relative decrement of network efficiency caused by the removal of a link. IC expects the link locating between communities to have high information centrality and the link locating within a community to have low information centrality [28]. Each iteration of IC accomplishes two tasks: calculating the information centrality of each link and removing the link with the highest information centrality. Experimental results showed that IC is effective at discovering community structures when the communities are cohesively connected with each other [28].

2.4. Edge-Clustering Coefficient (CD) Algorithm. Radicchi et al. [9] proposed the CD algorithm to solve two problems. The first problem is the quantitative definition of community, and the other problem is the time-consuming nature of divisive algorithms. To solve the first problem, they introduced two alternative quantitative definitions of community. To solve the second problem, they suggested a local centrality index edge-clustering coefficient. Based on the edge-clustering coefficient, CD can remove multiple links at each iteration.

2.5. Closed Walks (CW) Algorithm. Yang et al. [11] proposed the CW algorithm and introduced closed walks as a local centrality index. CW considers the closed walks of orders three and four based on three convincing pieces of evidence. The first evidence comes from statistical data where, in complex networks, the proportion of the links that participated in closed walks of orders 3 and 4 reaches ninety percent [11]. The second evidence comes from the *three degrees of influence* property of sociological significance [29]. The third evidence comes from the property that information usually propagates along paths without repeated nodes. Experimental results showed that CW is an effective way to solve the double peak structure problem [11].

3. An Autonomous Divisive Algorithm for Community Detection

3.1. Motivation. In real-world networks, it is often easier to discriminate between internal links and external links than to recognize overlapping nodes [1]. Defining communities as sets of links rather than nodes may be a promising strategy to analyze networks with overlapping communities [30, 31]. Based on this idea, many community detection algorithms [32–36] aim to find the differences in the property of links to extract high-quality community structures from networks. Based on network topology information, this paper discusses the difference between the properties of internal links and external links. We introduce a concept of the weak link to locate external links. In addition, we introduced a new link-break strategy and an autonomous division, so that the proposed divisive algorithm is free from parameters, nontopological information, and definition of community.

3.2. Definition of Weak Link. Many real-world complex systems can be represented as a graph $G = (V, E)$. V is the set of nodes, E is the set of links, $|V| = n$, and $|E| = m$. Most community detection algorithms are based on the notion that a community should have more internal connections than

external connections [1, 2]. This notion skillfully generalizes the difference of the density distribution between internal links and external links. However, more properties of links are urgently needed to make divisive algorithms free from parameters, nontopological information, and community definitions.

First, to tell the difference between the properties of internal links and external links, as a baseline, we investigate the expected contribution of a node to its neighbors for spreading information. If a node can only get its neighbors' information, then the node will expect that its neighbors' contribution for spreading information is uniform. We define the expected contribution that node j made to its neighbor i for spreading information as

$$E_i^j = \frac{1}{d_i}, \tag{1}$$

where d_i is the degree of node i.

Second, we investigate the property of internal links. In a community, core members, hub members, and outlier members play different roles in spreading information [37]. *Core members* contribute greatly to spreading information inside communities; *hub members* serve as hubs for spreading information both inside and outside communities; *outlier members* prefer to receive information rather than send information. In a community, from core members to outlier members, the node's contribution for spreading information declines. Therefore, if node i and node j are two endpoints of an internal link and R_i^j is the real contribution of node j to its neighbor i for spreading information, we expect that $R_i^j \geq E_i^j$ and $R_j^i \leq E_j^i$ or $R_i^j \leq E_i^j$ and $R_j^i \geq E_j^i$.

Lastly, we investigate the property of external links. The biggest difference between the properties of internal links and external links is that external links connect different communities. As the two endpoints of an external link play an important role in spreading information between communities, we expect that both of the endpoints have a real contribution which is greater than the expected contribution. Hence, if node i and node j are the two endpoints of an external link, we expect that $R_i^j > E_i^j$ and $R_j^i > E_j^i$. We define the weak link as follows.

Definition 1 (weak link). A link l with two endpoints i and j is a weak link if $R_i^j > E_i^j$ and $R_j^i > E_j^i$.

3.3. Determination of Weak Link. To determine whether a link is a weak link, it is essential to quantify the real contribution of the two endpoints of a link for spreading information. Thus, we investigate the structure of the shortest path tree of each endpoint and introduce the shortest path coverage as a measure.

We use the shortest path coverage to estimate whether a node is at the edge of a potential community based on the following observations. There are three subgraphs in Figure 1. The graphs in Figures 1(b) and 1(c) are the isomorphic graphs of Figure 1(a). In Figures 1(b) and 1(c), the solid lines present the shortest path tree of nodes A and B. If we consider Figure 1(a) as a community, then node A is a core member. In Figure 1(a), there are eight links, and there are four links and six links in the shortest path tree in Figures 1(b) and 1(c). We can see that there are four links in Figure 1(b) and two links in Figure 1(c) which are presented as dashed lines making no contribution to the shortest path tree; besides, the length of the shortest path from the source node A is 4 and the length of the shortest path from the source node B is 6. From Figure 1, we can summarize that, in a community, a core member gets in touch more quickly with the other members than a less important member, and the depth of the shortest path tree of a core member is shorter than that of a less important member.

To calculate the shortest path coverage, we have to calculate the end-frequency and arrival-frequency. Definitions of end-frequency, arrival-frequency, and the shortest path coverage are shown in Definitions 2, 3, and 4. Examples of the calculation of the three concepts are shown in Figure 2.

Definition 2 (end-frequency). In the shortest path tree, the end-frequency of node V is the number of distinct shortest paths that start from source node S and end at node V. End-frequency is written as E.

Definition 3 (arrival-frequency). In the shortest path tree, the arrival-frequency of node V is the number of distinct shortest paths that start from source node S and arrive at node V. Arrival-frequency is written as A.

Definition 4 (shortest path coverage). In the shortest path tree, suppose that node V is a neighbor of source node S; the shortest path coverage of node V is the proportion of the arrival-frequency of node V to the sum of the end-frequency of all the reachable nodes of source node S. The shortest path coverage is written as C.

The calculation of end-frequency is a top-down process using breadth-first search in time $O(m)$. We show an example for calculating the end-frequency in Figure 2(a). The end-frequency of the source node is 1. In the shortest path tree, the end-frequency of a node is the sum of the end-frequency of all its parent nodes. For example, in Figure 2(a), there is one shortest path from node S to node 1 and one shortest path from node S to node 2, and then the end-frequency of node 3 is $1 + 1 = 2$. The end-frequency is formulated as

$$E_{\text{child}} = \begin{cases} 1 & \text{Parents} = \varnothing \\ \sum_{\text{parent} \in \text{Parents}} E_{\text{parent}} & \text{Parents} \neq \varnothing, \end{cases} \tag{2}$$

where "Parents" is the parent node set of node child, "parent" is a node in "Parents," E_{child} is the end-frequency of node child, and E_{parent} is the end-frequency of node parent.

The calculation of arrival-frequency is a bottom-up process in time $O(m)$. We show an example for calculating the arrival-frequency in Figure 2(b). The arrival-frequency of a leaf node is its end-frequency. In the shortest path tree, the arrival-frequency of a node is its end-frequency plus the sum of its contribution to the arrival-frequency of its child nodes. In Figure 2(a), the end-frequency of nodes 2, 3, and 4 is 1,

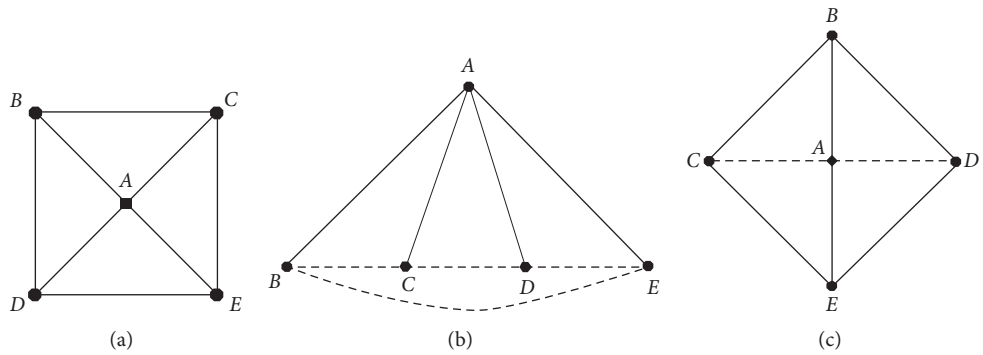

Figure 1: Observations of the shortest path tree.

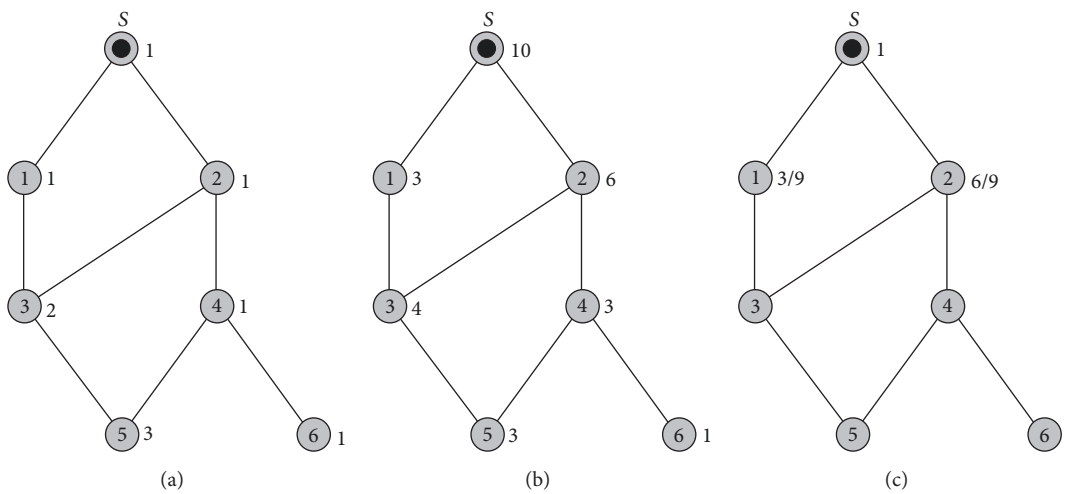

Figure 2: Calculation of end-frequency, arrival-frequency, and the shortest path coverage; the number inside the node is the node's ID. (a) The calculation of end-frequency; the number beside the node is the node's end-frequency; (b) the calculation of arrival-frequency; the number beside the node is the node's arrival-frequency; (c) the calculation of the shortest path coverage; the number beside the node is the node's shortest path coverage.

2, and 1, respectively. In Figure 2(b), the arrival-frequency of nodes 3 and 4 is 4 and 3. The contribution of node 2 to the arrival-frequency of nodes 3 and 4 is $4 \times 1/2 = 2$ and $3 \times 1/1 = 3$. In Figure 2(b), the arrival-frequency of node 2 is $1 + 2 + 3 = 6$. The arrival-frequency is formulated as

$$A_{\text{parent}} = \begin{cases} E_{\text{parent}} & \text{Children} = \varnothing \\ E_{\text{parent}} + \sum_{\text{child} \in \text{Children}} \left(A_{\text{child}} \dfrac{E_{\text{parent}}}{E_{\text{child}}} \right) & \text{Children} \neq \varnothing, \end{cases} \tag{3}$$

where "Children" is the child node set of node parent, "child" is a node in "Children," E_{parent} is the end-frequency of node parent, E_{child} is the end-frequency of node child, A_{parent} is the arrival-frequency of node parent, and A_{child} is the arrival-frequency of node child.

The shortest path coverage can be calculated in time $O(1)$. We show an example for calculating the shortest path coverage in Figure 2(c). For example, in Figure 2(b), the arrival-frequency of nodes 1 and 2 is 3 and 6; then, in Figure 2(c), the shortest path coverage of nodes 1 and 2 is $3/(3 + 6) = 1/3$ and $6/(3 + 6) = 2/3$. The real contribution of node j to its neighbor i for spreading information is given as

$$R_i^j = C_i^j = \frac{A_j}{\sum_{k \in \text{Neighbors}} A_k} = \frac{A_j}{(A_i - 1)}, \tag{4}$$

where "Neighbors" is the neighbor set of node i, k is a node in "Neighbors," A_j is the arrival-frequency of node j, and C_i^j is the shortest path coverage of j.

3.4. Autonomous Division and Link-Break Strategy. As shown in Section 2, several advanced algorithms have been proposed to detect communities in networks, but they all have certain limitations. For example, GN [25] and IC [28] are time-consuming on large-scale networks; DD [8] depends on some parameters; CD [9] and CW [11] depend on the order of cyclic structures. Besides, all these algorithms have a common limitation that the output of these algorithms depends on quality function or community definition. We proposed link-break strategy and autonomous division to overcome these limitations.

(1) **Input**: Graph G, node set V, link set E.
(2) **Output**: Weak link set W.
(3) **Process**:
(4) **for** each link $l_{ij} \in E$
(5) calculate E_i^j, R_i^j, E_j^i, R_j^i.
(6) **if** $R_i^j > E_i^j$ & $R_j^i > E_j^i$
(7) $W = W \cup l_{ij}$.
(8) **end if**
(9) **end for**
(10) **return** Weak link set W.

Algorithm 1: Determination of the weak link.

(1) **Input**: Graph G, node set V, link set E.
(2) **Output**: Community set C.
(3) **Process**:
(4) **Get weak link set** W from G.
(5) **while** $W \neq \varnothing$ **do**
(6) $E = E - W$, $G = \{V, E\}$, $W = \varnothing$.
(7) **Get weak link set** W from G.
(8) **end while**
(9) Each component is considered as a community.
(10) **return** community set C.

Algorithm 2: The AD algorithm.

To overcome the limitation on efficiency, a link-break strategy should have the ability to detect and remove multiple links at each iteration. To overcome the limitation on parameters and nontopological information, an autonomous division should take full advantage of the topology of the network. To overcome the limitation on quality function and community definition, an autonomous division should be able to terminate the algorithm when a satisfactory solution is reached.

The proposed link-break strategy is designed similarly to a tug-of-war contest. In the contest, communities act as contestants and links act as ropes. Weak links play the role of breakable ropes. When the force exerted on a breakable rope exceeds the rope's limit, the rope breaks. Then, the force exerted on the other ropes changes and other breakable ropes will continue to break. This process will repeat until there are no breakable ropes available in the system. Lastly, different communities get disconnected from each other.

Based on the weak link, autonomous division is easy to carry out. First, the concept of the weak link is proposed based on the topological properties of networks with a community structure. Second, based on the weak link, an algorithm has the ability to detect and remove multiple links at each iteration.

3.5. The Proposed Algorithm. Based on the concepts of the weak link and autonomous division, the proposed algorithm repeats detecting and removing weak links, until no weak links are left in the network. We show the determination of the weak link in Algorithm 1. We show the AD algorithm in Algorithm 2.

3.6. Time Complexity Analysis. Suppose AD algorithm works on a network with n nodes and m links. Based on the analysis in Section 3.3, at each iteration of the AD algorithm, the time complexity of the calculation of the shortest path coverage is $O(nm)$. Suppose that the number of potential weak links is n_{weak} and the number of iterations is I. Because at each iteration of Algorithm 2 multiple weak links can be removed from the network, according to step (5) to step (8), it can be inferred that $0 \leq I \leq n_{\text{weak}}$. In most free-scale networks, $I \ll n_{\text{weak}} \ll n$, so the time complexity of AD algorithm is $O(nm)$. In sparse graph which has an obvious community structure, the time complexity of the AD algorithm is $O(n^2)$. We list the time complexity of the AD algorithm and the other five divisive algorithms mentioned in Section 2 in Table 1.

4. Experiments and Results

In this section, the effectiveness of the AD algorithm is compared with the other five divisive algorithms mentioned in Section 2 on both artificial and real-world networks. All the experiments are conducted on a computer with Intel(R) Core(TM) i3 CPU, 2.66 GHz, and 2 GB RAM.

4.1. Evaluation Criteria

4.1.1. NMI. The normalized mutual information (NMI) is a similarity measure proven by Danon et al. [38]. NMI is based on defining a confusion matrix $\mathbf{N}$, where the rows represent real communities and the columns represent detected communities. N_{ij} is the element of $\mathbf{N}$, which represents the number of nodes that belong to real community i and detected community j. $N_{i.}$ is the sum of elements in row i, and $N_{.i}$ is the sum of elements in column i. Based on information theory, a measure of similarity between the partitions is then

$$I(R,F) = \frac{-2\sum_{i=1}^{c_R}\sum_{j=1}^{c_F} N_{ij}\log\left(N_{ij}N/N_{i.}N_{.j}\right)}{\sum_{i=1}^{c_R} N_{i.}\log\left(N_{i.}/N\right) + \sum_{j=1}^{c_F} N_{.j}\log\left(N_{.j}/N\right)}, \quad (5)$$

where $I(R,F)$ is the normalized mutual information, R represents the real partition, F represents the found partition, c_R is the real communities in R, and c_F is the detected communities in F. If the detected communities are identical to the real communities, then $I(R,F) = 1$. If the detected communities are totally independent of the real communities, then $I(R,F) = 0$.

4.1.2. Modularity. Girvan and Newman [10, 25] proposed modularity (Q) which is defined as

$$Q = \sum_{C=1}^{k}\left[\frac{l_C}{m} - \left(\frac{d_C}{2m}\right)^2\right], \quad (6)$$

where k is the number of detected communities, C is the ID of community, l_C is the number of internal links of C,

Table 1: Time complexity of divisive algorithms for community detection.

Algorithms	Time complexity	References
AD	$O(nm)$	[—]
GN	$O(nm^2)$	[10, 25]
DD	$O(n^3)$	[8]
IC	$O(nm^3)$	[28]
CD	$O(km)$	[9]
CW	$O(am + bm^2)$	[11]

Table 2: The parameters of artificial networks.

Networks	n	k	$\max k$	$\min c$	$\max c$	μ
LFR_μ	128	8	16	10	50	0.1~0.8
LFR_k	128	4~10	16	10	50	0.2

and d_C is the sum of the degrees of the nodes within C. This quality function measures the fraction of the links in the network that connect nodes of the same type minus the expected value of the same quantity in a network with the same community divisions but random connections between the nodes [10]. $Q = 0$ indicates that the number of links within the communities is only random. $Q = 1$ indicates the network with strong community structure.

4.1.3. I-Measure. In this paper, we use I to evaluate the division efficiency of the algorithms.

$$I = \text{The number of iterations of the algorithm.} \quad (7)$$

4.2. Data Sets

4.2.1. Artificial Networks. Lancichinetti-Fortunato-Radicchi (LFR) benchmark [39] produces networks with properties close to real-world networks. We use the LFR benchmark networks to test the algorithms. Some important parameters of the benchmark networks are given in Table 2. In Table 2, n denotes the number of nodes, k denotes the mean degree of the network, $\max k$ denotes the maximum degree of node, $\min c$ denotes the minimum size of community, $\max c$ denotes the maximum size of community, and μ denotes the mixing parameter. For LFR_μ, μ ranges from 0.1 to 0.8 with a span of 0.1. For LFR_k, k ranges from 4 to 10 with a span of 1.

4.2.2. Real-World Networks. The network of karate club *(Karate)* is a network of friendships between the 34 members of a karate club at a US university described by Zachary [40] in 1977. Zachary identified two communities of friendship in the network as shown in Figure 3.

The network of bottlenose dolphins *(Dolphins)* is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound compiled by Lusseau et al. [41]. A link between two dolphins was established by observation of the statistically significant frequent association. The network comprises two communities as shown in Figure 4.

The network of political books *(Books)* was compiled by Krebs [42]. The nodes represent 105 books on American politics brought from https://Amazon.com. 441 links join pairs of books frequently purchased by the same buyer. The network is composed of three communities as shown in Figure 5.

The network of American football games *(Football)* between Division IA colleges during regular season Fall 2000 was compiled by Girvan and Newman [25]. The network is composed of 11 conferences plus a few other teams without a clear affiliation as shown in Figure 6.

4.3. Experiment Results. In our experiments, we ignore any quantitative definition of community and achieve the partition when Q gets the maximum value. This will make the CD get better results, while reducing the efficiency. Besides, to avoid the local adjustment process of distance dissimilarity algorithm, DD removes the links that have the highest dissimilarity value at each iteration. We note that CD3 and CD4 represent the edge-clustering coefficient (CD) algorithm in orders 3 and 4.

4.3.1. Results on Artificial Networks. Figure 7 shows the results of the algorithms on LFR_μ data sets. The NMI values got by AD are about 0.15 lower than the average of the other algorithms. The Q values got by AD are close to the average of the other algorithms. Figure 8 shows the results of the algorithms on LFR_k data sets. When k is low, the NMI and Q values got by AD are lower than those got by the other algorithms. However, when k increases, the NMI and Q values got by AD explode, which means AD is more effective in discovering community structures when the communities are cohesively connected with each other.

From Figures 7 and 8, it seems that AD does not perform better than most of the other algorithms. Actually, all the other algorithms except for AD are guided by modularity Q as mentioned in Section 4.3, paragraph 1, which means Figures 7 and 8 show the best performance of the other algorithms. However, AD is not guided by modularity Q or any of the parameters, which means Figures 7 and 8 show the average performance of AD. Thus, we cannot say that AD performs worse than the other algorithms.

From Figures 7 and 8, we can observe that the I values got by AD are lower than those got by the other algorithms, which means that the link-break strategy of AD can reduce the number of iterations of divisive algorithm, thus improving the efficiency of the algorithm. Besides, we can observe that IC has the highest time complexity, which verifies the analysis of Table 1. Based on the *time cost* values got by the algorithms, we arrange the algorithms in an ascending order of time complexity: CD3 < CD4 < DD < CW < AD < GN < IC.

4.3.2. Results on Real-World Networks. From Table 3, we can observe that, for *Karate*, AD gets the highest NMI value. Besides, AD also gets a higher Q value than that of DD, CD4, and CW. For I-measure, AD algorithm gets the lowest I value.

From Table 4, we can observe that, for *Dolphins*, AD gets a higher NMI value than that of GN, IC, CD3, CD4, and

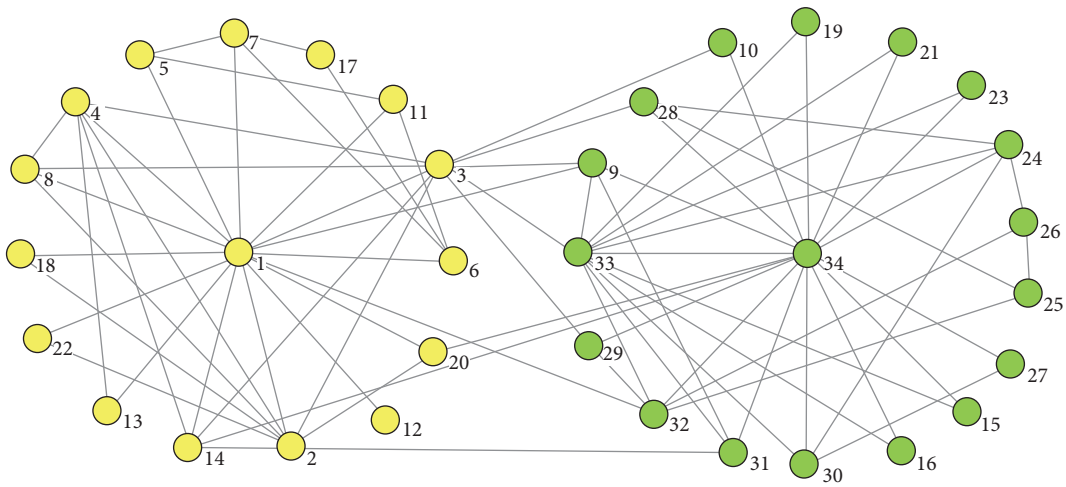

Figure 3: The network of the karate club.

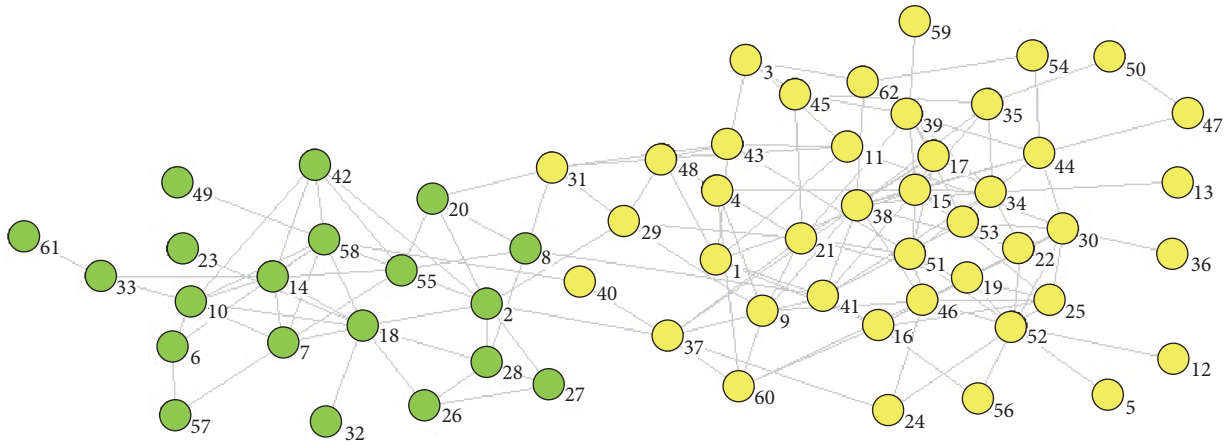

Figure 4: The network of the bottlenose dolphins.

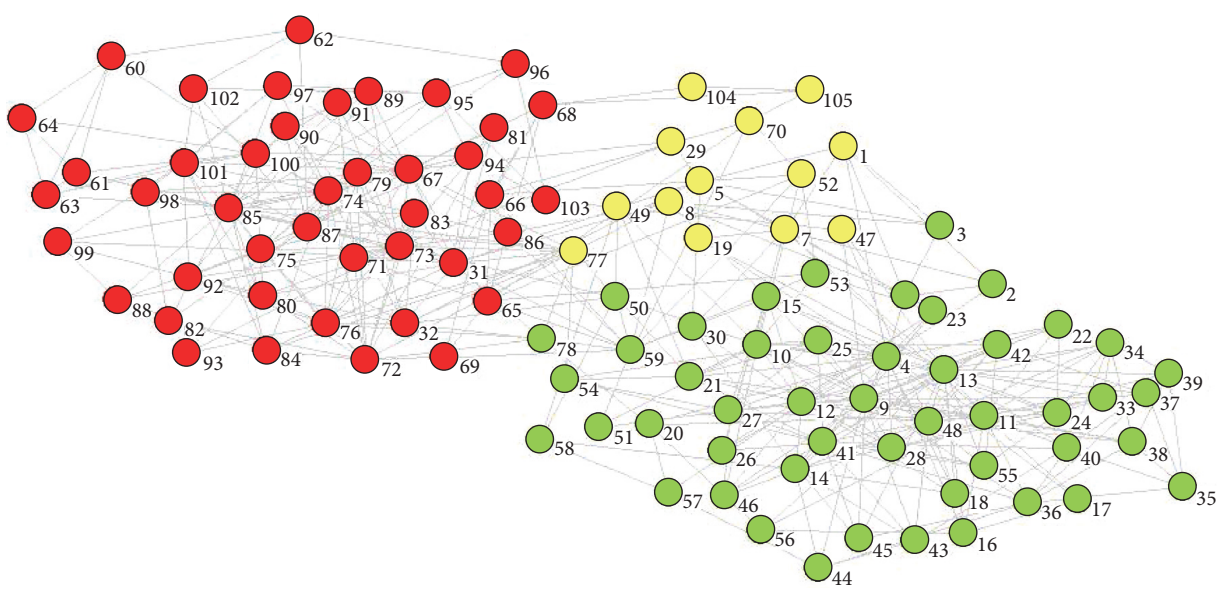

Figure 5: The network of the political books.

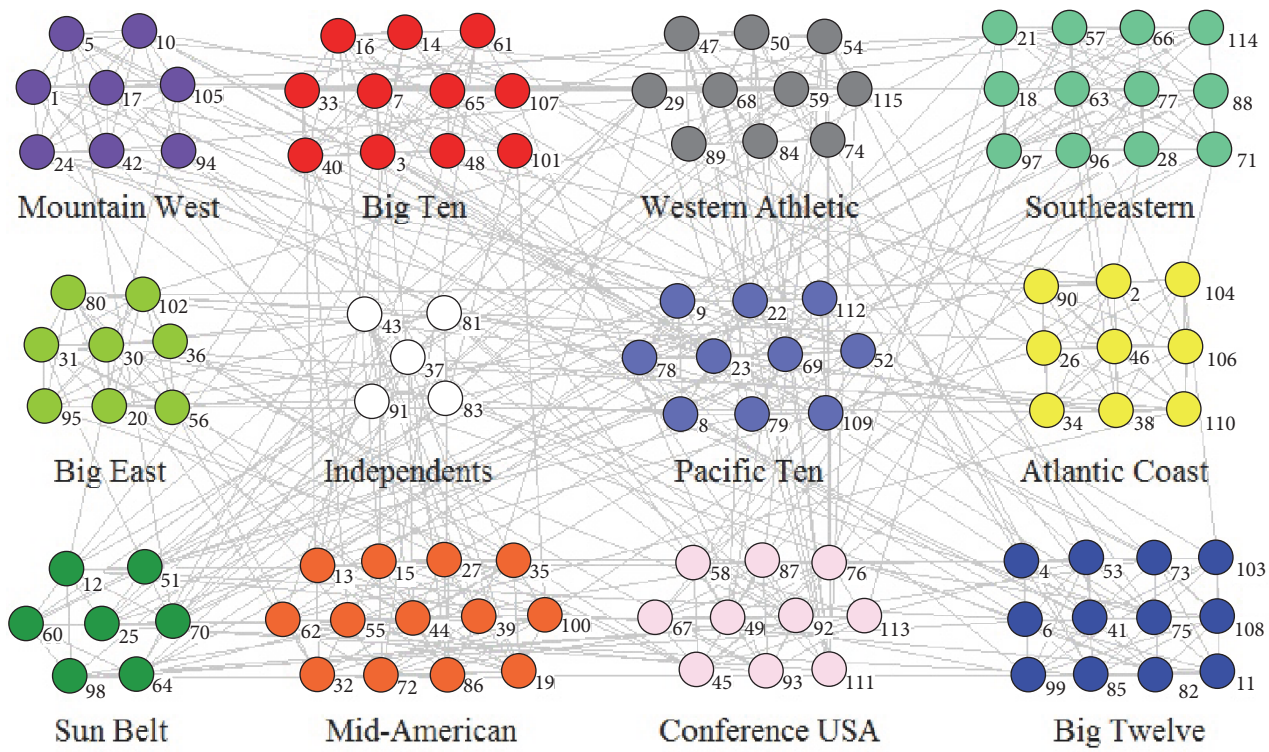

Figure 6: The network of the American football games.

Table 3: The results on *Karate*.

Measures	AD	GN	DD	IC	CD3	CD4	CW
NMI	1.00	0.58	0.39	0.66	0.67	0.57	0.64
Q	0.37	0.40	0.32	0.37	0.38	0.30	0.33
I	2	24	42	22	7	21	17

Table 4: The results on *Dolphins*.

Measures	AD	GN	DD	IC	CD3	CD4	CW
NMI	0.75	0.55	0.80	0.39	0.23	0.54	0.54
Q	0.39	0.52	0.38	0.45	0.41	0.50	0.52
I	3	32	19	54	17	22	19

Table 5: The results on *Books*.

Measures	AD	GN	DD	IC	CD3	CD4	CW
NMI	0.55	0.54	0.57	0.56	0.49	0.55	0.51
Q	0.52	0.52	0.49	0.51	0.52	0.48	0.49
I	4	51	66	45	84	19	40

CW. Besides, AD gets a higher Q value than that of DD. For I-measure, AD gets the lowest I value. From NMI and Q values in Table 4, we can also observe that NMI and Q are independent of each other. NMI is used to evaluate the quality of a partition when the real community structure is known, while Q is used to evaluate the quality of a partition when the real community structure is unknown.

Table 6: The results on *Football*.

Measures	AD	GN	DD	IC	CD3	CD4	CW
NMI	0.79	0.90	0.86	0.86	0.92	0.92	0.92
Q	0.58	0.60	0.60	0.60	0.60	0.60	0.60
I	5	186	189	177	29	94	91

From Table 5, we can observe that, for *Books*, AD gets a higher NMI value than that of GN, CD3, and CW. Besides, AD gets a higher Q value than that of DD, IC, CD4, and CW. For I-measure, AD gets the lowest I value.

From Table 6, we can observe that, for *Football*, AD gets the lowest NMI and Q value. There are two reasons for the poor results of NMI and Q. First, there are few teams without a clear affiliation. As shown in Figure 6, for the teams of conference "Independents," only teams 81 and 83 connected to each other. Second, some teams are more tightly connected with the teams from other conferences than the teams from the same conference. For example, all the teams of "Sun Belt" have more connections to the teams outside the conference than to the teams inside the conference. For I-measure, AD gets the lowest I value.

From Tables 3, 4, 5, and 6, we can observe that AD performs better in identifying communities from real-world networks than identifying communities from artificial networks. There are two reasons for this phenomenon. First, AD is proposed based on the differences between the properties of internal links and external links in the real-world networks

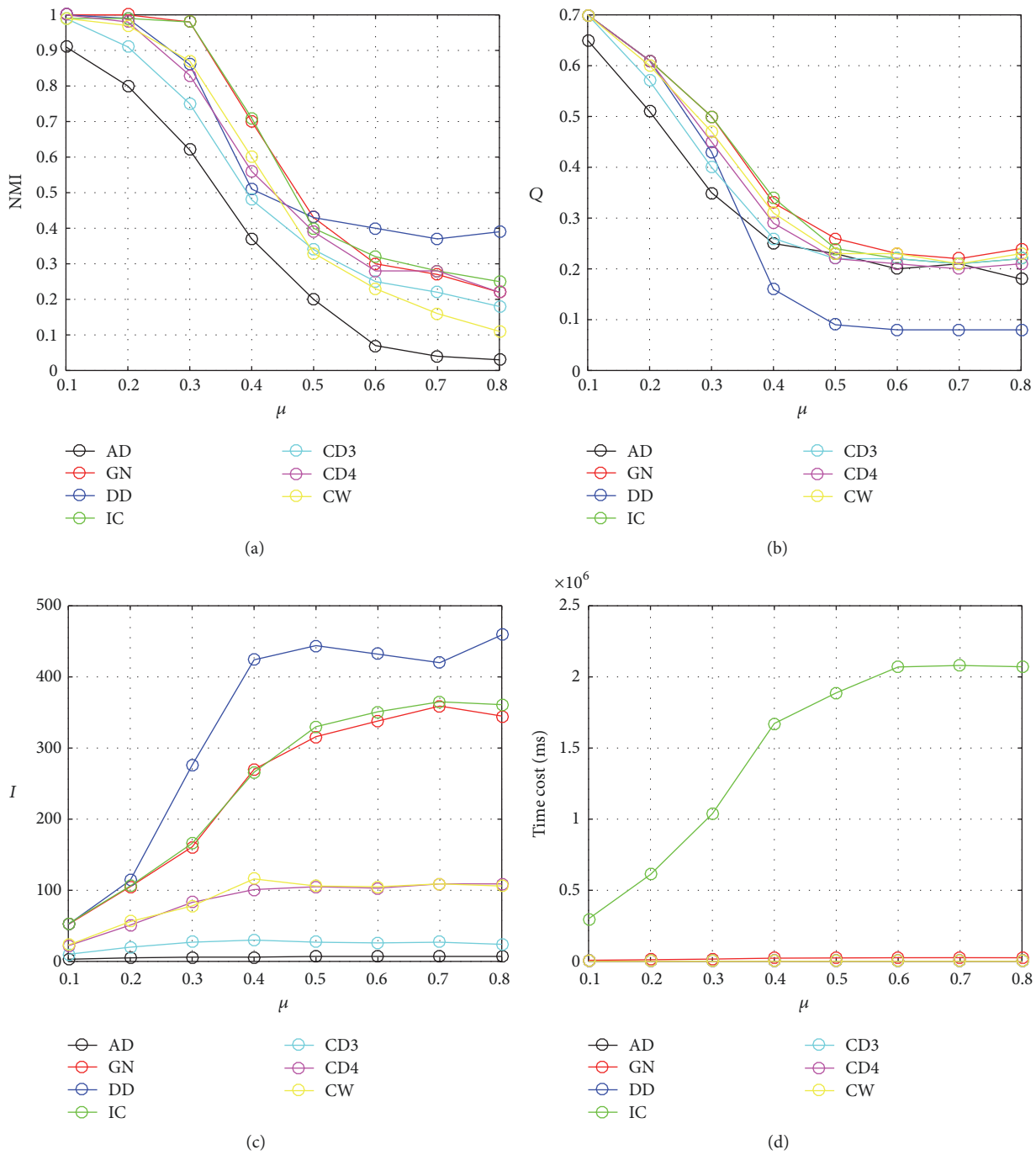

Figure 7: Results on LFR_{μ}.

where the internal and external links exhibit different characteristics. Second, LFR benchmark simulates some features of real-world networks (the node degree and community size are in power distribution); however, it does not consider the differences between the properties of internal links and external links. Therefore, we have a reason to believe that AD performs better in identifying communities from real-world networks than identifying communities from artificial networks.

5. Conclusions

In this paper, we proposed a new divisive algorithm to overcome the limitations on parameters, nontopological information, division efficiency, and community definitions. To make our algorithm free from parameters and nontopological information, we proposed the weak link which helps detect the links connecting different communities. To improve division efficiency, we proposed a link-break strategy based on the weak link, so that our algorithm could remove

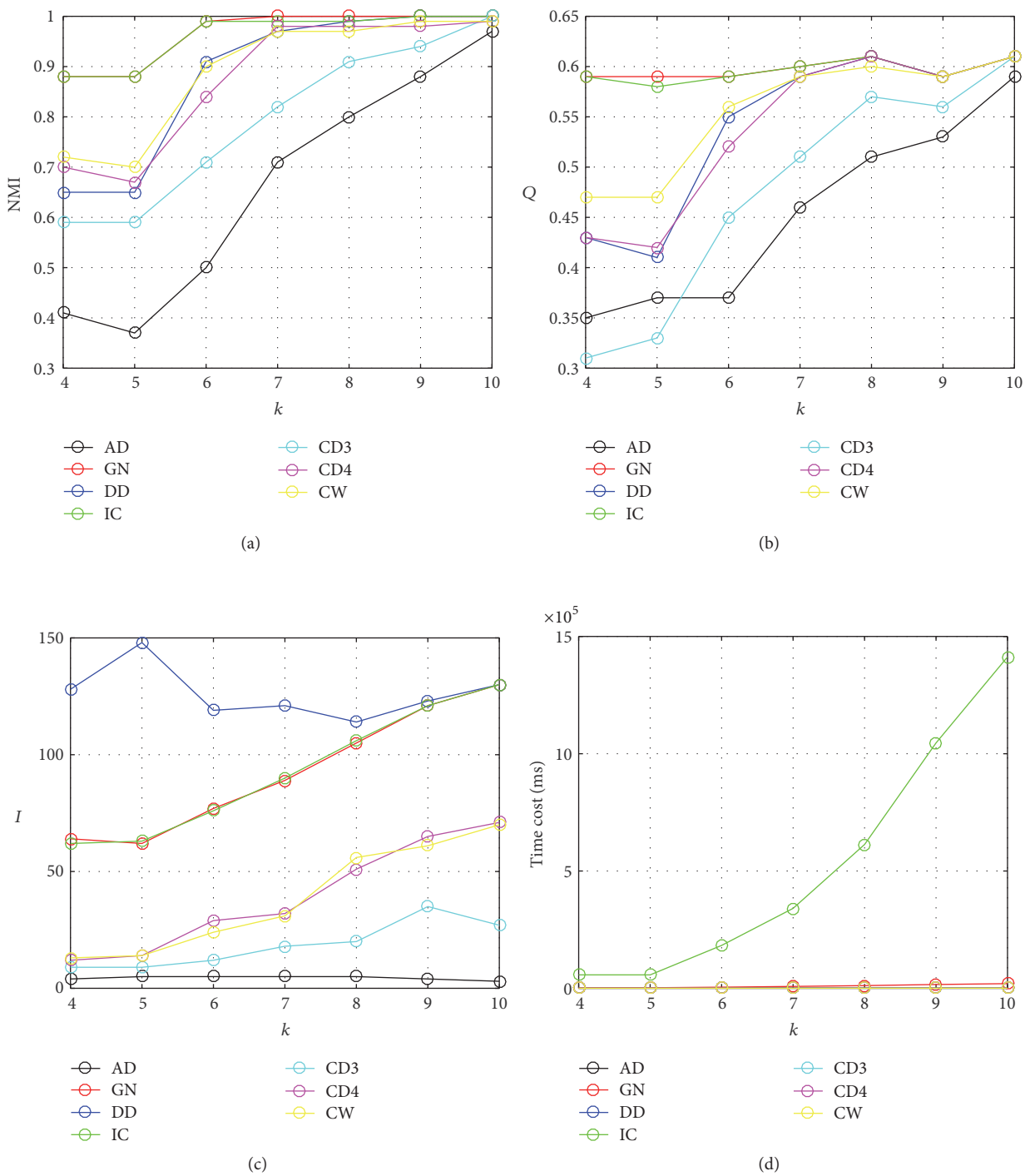

Figure 8: Results on LFR_k.

multiple links at each iteration. To overcome the limitation on community definition, we introduced an autonomous division in our algorithm to end the algorithm without the help of community definitions. Empirical evaluations on artificial and real-world networks showed that the proposed algorithm achieves a better accuracy-efficiency trade-off than some of the latest divisive algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by (1) the National Natural Science Foundation of China (nos. 61672179, 61370083, and

61402126), (2) Heilongjiang Province Natural Science Foundation (no. F2015030), (3) Province in Heilongjiang Outstanding Youth Science Fund (no. QC2016083), and (4) Heilongjiang Postdoctoral Fund to Pursue Scientific Research in Heilongjiang Province (no. LBH-Z14071).

References

[1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[2] 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, Article ID 2501657, 2013.

[3] W. Liu, M. Pellegrini, and X. Wang, "Detecting communities based on network topology," *Scientific Reports*, vol. 4, article no. 5739, 2014.

[4] X. Qi, W. Tang, Y. Wu, G. Guo, E. Fuller, and C. Zhang, "Optimal local community detection in social networks based on density drop of subgraphs," *Pattern Recognition Letters*, vol. 36, pp. 46–53, 2014.

[5] D. Rafailidis, E. Constantinou, and Y. Manolopoulos, "Landmark selection for spectral clustering based on Weighted PageRank," *Future Generation Computer Systems*, vol. 68, pp. 465–472, 2017.

[6] A. Mahmood, M. Small, S. A. Al-Maadeed, and N. Rajpoot, "Using geodesic space density gradients for network community detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 921–935, 2017.

[7] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, 2017.

[8] H. Zhou, "Distance, dissimilarity index, and network community structure," *Physical review. E, Statistical, Nonlinear, And Soft Matter Physics*, vol. 67, no. 6, Article ID 061901, 2003.

[9] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Paris, "Defining and identifying communities in networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2658–2663, 2004.

[10] 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, 2004.

[11] Y. Yang, P. G. Sun, X. Hu, and Z. J. Li, "Closed walks for community detection," *Physica A: Statistical Mechanics and its Applications*, vol. 397, no. 37, pp. 129–143, 2014.

[12] J. Q. Jiang, A. W. M. Dress, and G. Yang, "A spectral clustering-based framework for detecting community structures in complex networks," *Applied Mathematics Letters*, vol. 22, no. 9, pp. 1479–1482, 2009.

[13] L. Huang, R. Li, H. Chen, X. Gu, K. Wen, and Y. Li, "Detecting network communities using regularized spectral clustering algorithm," *Artificial Intelligence Review*, vol. 41, no. 4, pp. 579–594, 2014.

[14] L. Bai, X. Cheng, J. Liang, and Y. Guo, "Fast graph clustering with a new description model for community detection," *Information Sciences*, vol. 388-389, pp. 37–47, 2017.

[15] Z. Bu, G. Gao, H.-J. Li, and J. Cao, "CAMAS: A cluster-aware multiagent system for attributed graph clustering," *Information Fusion*, vol. 37, pp. 10–21, 2017.

[16] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, no. 6, Article ID 066111, 2004.

[17] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[18] J. Mei, S. He, G. Shi, Z. Wang, and W. Li, "Revealing network communities through modularity maximization by a contraction-dilation method," *New Journal of Physics* , vol. 11, Article ID 043025, 2009.

[19] J. Xiang, T. Hu, Y. Zhang et al., "Local modularity for community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 443, pp. 451–459, 2015.

[20] D. Gómez, J. T. Rodríguez, J. Yáñez, and J. Montero, "A new modularity measure for fuzzy community detection problems based on overlap and grouping functions," *International Journal of Approximate Reasoning*, vol. 74, pp. 88–107, 2016.

[21] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, Article ID 036106, 2007.

[22] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics* , vol. 12, Article ID 103018, 2010.

[23] H. Sun, J. Liu, J. Huang et al., "CenLP: A centrality-based label propagation algorithm for community detection in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 436, pp. 767–780, 2015.

[24] R. Francisquini, V. Rosset, and M. C. V. Nascimento, "GA-LP: A genetic algorithm based on Label Propagation to detect communities in directed networks," *Expert Systems with Applications*, vol. 74, pp. 127–138, 2017.

[25] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[26] M. E. Newman, "Mixing patterns in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 67, no. 2, Article ID 026126, 2003.

[27] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 2004.

[28] S. Fortunato, V. Latora, and M. Marchiori, "Method to find community structures based on information centrality," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 70, no. 5, Article ID 056104, 2004.

[29] N. A. Christakis and J. H. Fowler, "Social contagion theory: examining dynamic social networks and human behavior," *Statistics in Medicine*, vol. 32, no. 4, pp. 556–577, 2013.

[30] T. S. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 1, Article ID 016105, pp. 145–148, 2009.

[31] Y. Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[32] J. Cheng, M. Leng, L. Li, H. Zhou, and X. Chen, "Active semi-supervised community detection based on must-link and cannot-link constraints," *PLoS ONE*, vol. 9, no. 10, Article ID e110088, 2014.

[33] D. Deritei, Z. I. Lázár, I. Papp et al., "Community detection by graph Voronoi diagrams," *New Journal of Physics* , vol. 16, Article ID 063007, 2014.

[34] Y. Chen, X. L. Wang, B. Yuan, and B. Z. Tang, "Overlapping community detection in networks with positive and negative links," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 3, Article ID P03021, 2014.

[35] Y. Xu, H. Xu, and D. Zhang, "A novel disjoint community detection algorithm for social networks based on backbone degree and expansion," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8349–8360, 2015.

[36] L. Liu, W. L. Zuo, and T. Peng, "Detecting outlier pairs in complex network based on link structure and semantic relationship," *Expert Systems with Applications*, vol. 69, pp. 40–49, 2017.

[37] H. Zhuge and J. Zhang, "Topological centrality and its e-Science applications," *Journal of the Association for Information Science and Technology*, vol. 61, no. 9, pp. 1824–1841, 2010.

[38] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, Article ID P09008, 2005.

[39] 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.

[40] 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.

[41] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait?" *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[42] V. Krebs, *Social network of political books*, 2004, http://www.visualcomplexity.com.