information

MDPI

Article

Community Detection Based on a Preferential Decision Model

Jinfang Sheng, Ben Lu, Bin Wang *, Jie Hu, Kai Wang, Xiaoxia Pan, Qiangqiang Dong and Dawit Aklilu

School of Computer Science and Engineering, Central South University, Changsha 410083, China;
jfsheng@csu.edu.cn (J.S.); itachi@csu.edu.cn (B.L.); 174712017@csu.edu.cn (J.H.); wangkaicsu@csu.edu.cn (K.W.);
1402140114@csu.edu.cn (X.P.); 174612241@csu.edu.cn (Q.D.); dawa.dejene@gmail.com (D.A.)
* Correspondence: wb_csut@csu.edu.cn

Received: 11 December 2019; Accepted: 14 January 2020; Published: 18 January 2020

Abstract: The research on complex networks is a hot topic in many fields, among which community detection is a complex and meaningful process, which plays an important role in researching the characteristics of complex networks. Community structure is a common feature in the network. Given a graph, the process of uncovering its community structure is called community detection. Many community detection algorithms from different perspectives have been proposed. Achieving stable and accurate community division is still a non-trivial task due to the difficulty of setting specific parameters, high randomness and lack of ground-truth information. In this paper, we explore a new decision-making method through real-life communication and propose a preferential decision model based on dynamic relationships applied to dynamic systems. We apply this model to the label propagation algorithm and present a **C**ommunity **D**etection based on **P**referential **D**ecision Model, called **CDPD**. This model intuitively aims to reveal the topological structure and the hierarchical structure between networks. By analyzing the structural characteristics of complex networks and mining the tightness between nodes, the priority of neighbor nodes is chosen to perform the required preferential decision, and finally the information in the system reaches a stable state. In the experiments, through the comparison of eight comparison algorithms, we verified the performance of CDPD in real-world networks and synthetic networks. The results show that CDPD not only has better performance than most recent algorithms on most datasets, but it is also more suitable for many community networks with ambiguous structure, especially sparse networks.

Keywords: complex networks; community detection; preferential decision; label propagation; information dynamic

1. Introduction

Today, great progress has been made in the research of complex networks. Complex network theory is widely used in social, economic, biological, transportation, and other fields. The nodes and edges in the network can truly reflect the existing relationships and the invisible relationships that may exist in the future. For example, the Theory of Six Degrees tells us that two seemingly unrelated people need only six people to get connected [1]. With the structure of complex network becoming more and more complicated, some characteristics of the network have aroused a widespread concern of related scholars [2]. A very important part of a complex network is the community structure of complex network. The connections between nodes are different [3]. Some nodes are closely connected, others are sparse, and even one node is independent. These can all be defined as associations. The general concept of a community is that the connections between members of the community are closer than those outside the community [4]. Therefore, a common community detection method is mainly to calculate the closeness of these members through a certain algorithm to realize the division of the community networks.

Thus far, exploring community structure in complex networks has attracted great attention. Community structures that are also called groups, clusters, cohesive subgroups or modules in different contexts are an important research direction in complex networks. As an important character of complex networks, the division of community structure can not only deeply understand the function of each community in the network and the relationship with the whole network function, but also explore the relationship between different communities in the network, which is helpful in understanding the network topology [5]. A large number of community detection algorithms based on various theories have emerged in the past years. All algorithms have their own advantages and shortcomings. For example, some algorithms have short time complexity but may require parameters. Some algorithms are stable, but their accuracy is poor. Hence, with the increasing complexity of the network, accurate and efficient partitioning in complex networks is still a hot research direction, which attracts numerous people to confront the challenge.

1.1. Basic Idea

The research of complex networks is a hot issue in many fields. Among them, community detection is a complex and meaningful process that plays an important role in studying the characteristics of complex networks. We regard the topological network as a dynamic circle of life that is constantly communicated in real life [6], and nodes communicate with each other through learning behaviors to make decisions.

In the process of information exchange in real life, learning strategies and learning behaviors of people are easily affected by the surrounding environment. When we make decisions, we are often affected by the environment at the time. It is not difficult to understand that we tend to choose to exchange information with people who are more influential, closer to us and more similar to our knowledge structure. Thus, we are influenced by the people around us when we make a decision. It helps us make decisions by learning from the experiences of others. There are many types of learning in different circles. Like those around us who have great resources and influence, and those who have a close relationship with ourselves, we will be greatly influenced by them when making decisions. This is also applicable to community discovery in complex networks. In the continuous iteration of local communities, nodes constantly make decisions, and eventually stabilize.

A preferential decision-making mechanism based on dynamic relationship mainly includes decision-making behaviors, learning strategies, and relationships. Information is exchanged through links between these three factors. The clustering process of the information dynamic system is described as follows: The first step is the node initialization process, giving each node in the network a unique label. The second step is the information iterative process. During the interaction of the nodes in the dynamic system [7], they will comprehensively consider the decision based on the information of the surrounding nodes. The last step is to uncover the community structure. Each node tends to a stable state, and nodes with the same information category will cluster into the same community.

A simple network with eight nodes can be represented in Figure 1. The color of the nodes represents the information carried by the nodes. The same color represents clustering into a community and the thicker the edges between the nodes, the better the priority between the nodes. Figure 1a displays information initialization. Each node is assigned its unique label information. At this stage, the nodes have the same degree of preference for surrounding nodes. Figure 1b shows the information dynamic interaction. Nodes continuously exchange information during the iterative process of the dynamic system. The tightness of the connection between the nodes has been changing during the dynamic iteration, and, through the introduction of the preferential decision mechanism, the system finally stabilized. Figure 1c displays uncover communities. It can be known that the preference relationship between nodes in the same community will be much larger, and the number of contact interactions will be too high. This is consistent with the definition of community in complex networks. Finally, the system achieves stable community division.

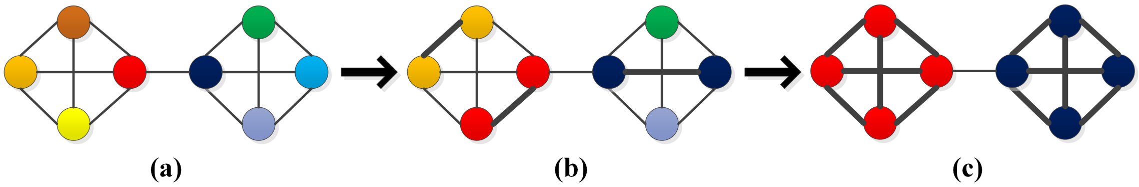

Figure 1. A sample on a simple network. (**a**) information initialization, (**b**) information dynamic interaction, (**c**) uncover communities.

1.2. Contributions

For traditional label propagation algorithms, the higher randomness and poor accuracy are common disadvantages. There are two main differences between our method and the existing ones in algorithm design. First, the existing algorithm learns the labels of surrounding neighbors in the process of label propagation randomly. Such random learning behavior is the main thing that leads to the large randomness of the algorithm results and the lower accuracy. In our method, a dynamic relationship-based preferential decision mechanism is introduced that can make the label learning behavior have a certain basis instead of random selection, which improves the stability and accuracy to some extent. Secondly, we know that the results of algorithms with parameters often depend on the setting of parameters. Our algorithm is non-parametric. Compared with other algorithms with parameters, our algorithm can automatically uncover communities without prior knowledge and parameter settings. Based on the improvement of such two-point algorithm design and a large number of experiments, we can give four advantages of our algorithm:

1. **Higher accuracy.** By comparing classic and latest algorithm experiments in the complex network, the CDPD algorithm has an accurate community partition on the real-world network and the synthetic network. In addition, the discovered communities are relatively close to the real communities.
2. **Stability.** The randomness has been greatly reduced by introducing the preferential decision mechanism and the past memory similarity.
3. **Scalability.** Although the iterative process based on dynamic relationships is complicated, CDPD only needs to consider the characteristics of neighboring nodes, which results in low time complexity of $O(t \times k^2 \times n)$. The values of t and k are usually small. Therefore, the CDPD algorithm can be applied to large-scale networks.
4. **Free parameters.** Instead of relying on prior knowledge and parameter setting methods, the CDPD method does not require parameter settings, and the communities are automatically discovered based on the local topology of the network.

2. Related Work

Community detection problems in complex networks have received extensive attention from academia since they were proposed. A large number of excellent algorithms have been published in recent years. These algorithms try to solve community detection problems from various angles. This section gives a brief summary of the current state of development regarding this issue from several unique perspectives. More detailed explanations and summaries can be referred from the excellent reviews [8–10].

Parameter Optimization Based Methods. Some algorithms discover the hidden community structure in a complex network by maximizing network characteristics or evaluating indicators. Usually, the features or parameters that can be selected are density [11], closeness [12] and modularity, and especially algorithms based on modularity optimization are emerging endlessly. The modularity

(Q) [13] is widely used as evaluating the quality of community detection. The higher the modularity value obtained by an algorithm is, the better the segmentation performance is. However, the modularity-based optimization problem has been proved to be an NP-complete problem [14], but there are still many algorithms that can achieve similar results through certain processing methods. Among them, the more representative ones are the FN algorithm [15] and the EO algorithm [16]. The common problem of this kind of algorithm is that the performance of clustering is not good, especially in some large real-world networks.

Game Theoretic Model Based Methods. The research of game theory provides a novel perspective for the problems of community detection. The game theory models mainly used in this problem include non-cooperative games and cooperative games [17–19]. In the non-cooperative game, nodes are considered to be selfish individuals. Each node guarantees that its benefit is maximized through the implementation of the strategy. When all nodes can't get more benefits by changing the strategy, we think that the system has reached a stable state, which is Nash equilibrium. The strategy here is usually to join, leave, and maintain existing community relationships. Common algorithm implementations include NASHDeCo algorithm [20], CaoGame [21], and DGT [22]. What is different from the non-cooperative game is that the nodes of the cooperative game model need to maximize the overall benefit by changing the strategy and dividing the community when the system reaches the Nash equilibrium. Some good algorithms are implemented along these lines, such as SNS-CD [2]. Nevertheless, challenges that such game theory-based algorithms cannot avoid are how to determine the utility function and how to define the equilibrium conditions, which are often difficult.

Dynamic System Based Methods. The dynamic-based community detection algorithm is another significant research route. This kind of method regards complex networks as a dynamic system and achieves the purpose of community detection through dynamic interaction. One of the angles is based on random walks. The algorithm represented WalkTrap(WT) [23]. The main idea of the WT is to construct a similarity evaluation method between nodes. It must be mentioned that WT's high time complexity ($O(mn^2)$) makes it unsuitable for large-scale networks. Another important research route is the label propagation algorithm (LPA) [24]. Although LPA has linear time complexity, one drawback that cannot be ignored is randomness. This comes from two aspects. On one hand, the node update order is random, on the other hand, when the neighbors have more than one maximum number of the labels, the node randomly selects one of the labels. Although some scholars have made improvements to LPA from various aspects [25–27], there is no algorithm that considers label number, label history, and other information comprehensively when selecting labels. There are also some superior dynamic system-based algorithms such as synchronization [28], distance dynamics [29], and so on.

3. Community Detection Based on Preferential Decision

3.1. Basic Formula Principle and Concepts

Before getting into the detail of our proposed algorithm, we first introduce some basic definitions that will be used in the following sections. All key notations are summarized in Table 1. Let $G = (V, E)$ be a given network, where V is the set of nodes and E is the set of edges.

Table 1. A summary of the notations.

Symbol	**Definition**
$\|V\|$	The number of nodes
$\|E\|$	The number of edges
$M_u[i]$	The i-th memory tag of the node u
$N(u)$	The neighborhood of node u
$J(u,v)$	The Jaccard similarity coefficient of node u and node v
$CS(u,v)$	The contact strength of node u and node v
$MS(u,v)$	The past memory similarity of node u and node v
$PD(u,v)$	The preferential decision degree of node u for node v
$P(u,v_k)$	Node u determines the probability of selecting node v

Definition 1 (Jaccard similarity coefficient). *It is used to compare similarities and differences between finite sample sets. The larger the Jaccard coefficient is, the higher the similarity of samples is. Given an undirected network,* $G = (V, E)$*, the Jaccard coefficient of node u and node v is defined as:*

$$J(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}, \tag{1}$$

where $\Gamma(u)$ *means a set of neighborhoods of node u that includes its adjacent nodes* $N(u)$*.* $\Gamma(u)$ *is defined as:*

$$\Gamma(u) = \{v \in V \mid \{u,v\} \in E\} \cup \{u\}. \tag{2}$$

Definition 2 (Contact strength). *There are different levels of connection strength between two nodes in any networks, which can be seen as good or bad in interpersonal relationships. This concept plays an important role in information exchange. We use the degree of the node as its initial information, and introduce the connection strength to indicate the connection tightness between nodes. Given an undirected network* $G = (V, E)$*, the contact strength of node v on node u is defined as:*

$$CS(u,v) = \frac{|N(u) \cap N(v)|}{T_u}, \tag{3}$$

where T_u *means the number of triangles owned by node u. The number of triangles shared by node u and node v is* $|N(x) \cap N(y)|$*.*

3.2. Preferential Decision Model Based on Dynamic Iteration

Based on the formulas and definitions mentioned above, a new model has been proposed, which is a preferential decision model based on dynamic iteration that involves many aspects: the information propagation model, the memory similarity, preferential decision degree, the loss of information, and so on.

In real life, we make progress through continuous learning. No matter what age, we will be influenced by the surrounding environment to make decisions, which imperceptibly drives us to learn from the people or things around us. We compare the community network to a social circle. When we make decisions, we will be influenced by the people around us. Learning from the experience of others to help us make decisions is the core idea presented in this article.

There are many types of learning in different circles. Like those around us who have resources and influence and those who have a close relationship with us, we will be greatly influenced by them when making decisions. This is also true for community detection in complex networks. In the constant iterations of local communities, nodes are constantly making decisions and finally approaching stability. In our daily life, we can understand that parents have a stronger connection with us. For example, their suggestions will have a great impact on the decision of which we will settle in the future. This is one of the reasons why we want to introduce connection strength. Compared with our own past behavioral cycle, it shows that our hobbies and life rhythms are very similar. Because different people have different decision-making styles, people with identical similarities will be more likely to get together. We are closer to making this decision, which is close to the life we are exposed to. Thus, the past similarity between the two nodes is introduced to reflect the true tightness of the two nodes. In each iteration, for neighbor nodes, each node will make its own decision according to the surrounding comprehensive factors. Thus, in the end, we lead to the probability that one node chooses another node, and there is a certain choice for any node around. Some of the probabilities may approach zero, in which case the node will basically not make its own choice. In the information propagation based on dynamic iteration, the neighboring nodes are de-spreaded with a certain probability. For example, in our real life, we will pass on information, ideas, attitudes or affections to other individuals or groups in order to make resultant changes. In information dissemination, there is a possibility of information loss, which is directed by the topological features and transmitted information. Because there are always some useless or uninteresting information in real life to interfere with us, we consider the information propagation model, the past memory similarity, preferential decision degree, and the loss of information together.

The past memory similarity. Given a list of memory tags to any social network node, the higher the similarity of the past is, the closer the past behavior of the node is. This shows that the probability of interaction between two nodes is greater. In the following formula, $M_u[i]$ means the i-th memory label list array of the node u. In particular, ∂ is a constant, usually 0.001, and the purpose of introducing ∂ is to prevent $ML(u,v)$ from falling to zero. We have done numerous experiments to change the value of ∂ but found that the performance is the best in this case. We can define $ML(u,v)$, which is the past memory similarity between node u and node v, as follows:

$$ML(u,v) = \partial + \textstyle\sum_{M_u[i]=M_v[i]} \frac{1}{2^i}. \tag{4}$$

We can identify memory labels as something that a person was interested in, and the length of the memory area represents the range of personal acceptance. The more similar the memory tags are, the greater the value of $ML(u,v)$ is, which means the more similar common experience between the two nodes. In real life, people with the same experience share common interests and hobbies. Because different people have different decision-making styles, people with similar experiences are more likely to get together.

Preferential decision degree. In each iteration of information propagation, each decision of the node is the result of preferential selection after comprehensive consideration. Thus, we introduce the concept of preferential decision-making degree to better represent the stability of the algorithm. When choosing community division, it is analogous to choosing to communicate with people around us in our daily life and make their own decisions. In real life, we prefer to go to communicate with people who are similar to past cycle behaviors. People who have similar interests and interests will hope to learn what they want from people who have close relationships with them. Generally, let $N(u)$ be the set of neighbors of node u, $CS(u,v)$ and$J(u,v)$ indicate the connection strength and the Jaccard similarity between node u and node v, respectively. $ML(u,v)$ denotes the past memory similarity coefficient between node u and node v. Let $PD(u,v)$ be the preferential decision degree, which is defined as follows:

$$PD(u,v) = |N(u)| \times |N(u)| \times J(u,v) \times e^{CS(u,v)+ML(u,v)}. \tag{5}$$

The propagation of information between nodes is analogous to the exchange of information among people. Communication between people can be influenced by many factors. Knowledge and resources of a person can be regarded as the Preferential decision degree of a node. The association between the two persons can be considered as the connection strength and the Jaccard coefficient. In the process of iteration, the node will always choose the best result to make a reasonable decision.

The probability of propagation. From the above formulas, we can conclude that the node has a preferential decision degree for each neighboring node, and the higher it is, the greater the decision-making probability will be. Let $N(u)$ be the set of neighboring nodes of node u, and node v_k belongs to $N(u)$. Let $P(u, v_k)$ be the probability of selecting node v_k as deciding objects for node u, which is defined as follows:

$$P(u, v_k) = \frac{PD(u, v_k)}{\sum_{v_k \in N(u)} PD(u, v_k)}. \tag{6}$$

For the node u, all its neighbors are likely to be selected. Low Preferential decision degree indicates that the probability of being selected is small, but this does not mean that it is impossible. It can also be analogous to our real life, and sometimes we need to learn what we need from strangers.

Information loss. In our daily lives, there is always some information that will be discarded in the process of information exchange. As the number of iterations increases, the probability of information loss increases. To prevent the information exchange failure caused by the preferential decision degree between nodes, it is extremely important to introduce a threshold. The copra algorithm divides the community by introducing a label dependent coefficient and an adjustable parameter v. In this paper, a threshold is used to control nodes to make better decisions. Its threshold has a significant impact on the experimental results, which not only reduces the reference of parameters but also divides the community accurately in a better way according to its node characteristics.

3.3. Community Detection Based on the Preferential Decision Algorithm

In this section, based on the above-mentioned content, we introduce the preferential decision algorithm. The proposed algorithm is similar to the traditional tag propagation algorithm. Whether it is a simple network or a complex network, it has its specific topological structure. Initially, the node does not have any interactive information behavior. We initialize each node's information and give each node a unique label to ensure its atomicity. In addition, the connection strength and the Jaccard coefficient between nodes are calculated, and the close relationship between nodes is analyzed. After the initialization of the previously mentioned information, it enters the dynamic iteration stage. Information is transmitted in the network, and the nodes interact with each other continuously. According to its decision-making probability and goal, it standardizes its own tag list. In each step, each node constantly updates its own memory tag list. Finally, with continuous iteration, due to the influence of topology-driven, all nodes in the network will reach a stable state, and the memory labels of nodes will also reach a convergent state. When the information in the network converges, the labels of the same community will naturally be the same, and the labels of different communities will be different. Therefore, after considering the comprehensive factors, we can naturally uncover the community, so as to achieve the performance of community division.

From the above, we can propose that the algorithm flow is similar to the traditional label propagation algorithm. It can be roughly divided into three categories: (a) information initialization, (b) information dynamic interaction, and (c) community detection. The CDPD algorithm is given in Algorithm 1 as follows:

Algorithm 1 CDPD

```
Input: $G = (V, E)$;
//Information initialization.
for each node $u$ in $V$ do
    Initialize label($u$) = $u$;
    Get the $N(u)$;
    for each node $v$ in $N(u)$ do
        Set the memory label;
        Compute the Jaccard similarity coefficient using Equation (1);
        Compute the connection strength using Equation (3);
    end for
end for
//Information dynamic interaction.
$Symbol = TRUE$
while $Symbol$ do
    Integrate order($V$);
    for each node $u$ in $V$ do
        for each node $v$ in $N(u)$ do
            Compute the past memory similarity (cf. Equation (4)) between $u$ and $v$;
            Compute the preferential decision degree (cf. Equation (5)) of $u$ for $v$;
            Learn and normalize labels;
            Decide the target of $u$ on the basis of probability of propagation;
            if satisfy the stop iteration condition of the algorithm then
                $Symbol = FALSE$;
            end if
        end for
    end for
end while
// Community detection.
for each node $u$ in $V$ do
    for each label $k$ in label list of node $u$ do
        if the label list of node $u$ is equivalent then
            Set the label of node $u$;
        else
            $u- > C_k$;
        end if
    end for
end for
// Return the resulting components C(communities)
Set $Set_c$=$\{C_1, C_2, C_3 ... C_n\}$ and $n$ is number of community.
Output: $C$.
```

3.4. Complexity Analysis

In the CDPD algorithm, we mainly analyze the time complexity from three parts. It is summarized as follows:

Information initialization. The initial state is labeled for each node, and a loop is only needed. Thus, its time complexity is $O(n)$. At the same time, our algorithm needs to calculate some additional necessary information, such as memory similarity, connection strength, Jaccard coefficient, and so on. The time complexity of this process is $O((k+1) \times n)$, where k is the average degree of the whole network.

Information dynamic interaction. In each iteration process, each node only chooses one node at one time as the target, and its time complexity is $O(|L| \times max)$, where $|L|$ is the length of its memory tag, and max is the maximum length of the tag in the network. In addition, the time complexity of deciding the target is $O(max)$. The overall number of iterations is denoted as t, which is typically between 30 and 100. Thus, the time complexity of this process is $O(t \times n \times (|L| \times max + max))$.

Community detection. The time complexity of this process is $O(num \times n)$, where num stands for the number of communities in the whole network, usually $num << n$.

In summary, the time complexity of the CDPD algorithm is the sum of the three parts: $O((k+1) \times n + t \times n \times (|L| \times max + max) + num \times n)$. We note that the values of t and max are usually small. Therefore, for small and sparse networks, our algorithm can be considered to be nearly linear.

4. Experiment

In this section, we first briefly introduce several common representative algorithms as the proposed comparison algorithm. Afterward, the datasets used in this experiment are introduced, which contain the real-world datasets and the synthetic datasets.Then, we select the appropriate evaluation metrics based on the characteristics of the datasets. Finally, our algorithm and comparison algorithm are applied to each dataset to observe and analyze the results of the experiment. To eliminate the randomness of the algorithm, each experiment was repeated 100 times to take the average as the experimental result. The experimental environment is i5-4590 3.2 GHz CPU, 8 GB RAM, Win8 OS PC.

4.1. Comparing Algorithms

To evaluate the performance of CDPD, we compare it with several representatives of community detection algorithms which belong to different types and different periods to verify the universality and the accuracy of CDPD. The basic principle and time complexity of all algorithms are shown in Table 2.

Fastgreedy. FG greedily maximizes the modularity of the graph by dividing nodes into communities [30]. The algorithm stops running when the modularity value of the graph is no longer increasing.

Spinglass. SG maps the community structure into the spin configuration and takes the spin state as the community indices to realize the community detection by minimizing the energy of the spin glass [31].

Infomap. In this algorithm, the probability flow in the graph theory is used to represent the information flow in the real-world network and then the community structure in the network is discovered through probability flow processing [32].

LPA. LPA is one of the fastest community detection algorithm based on label propagation implementation, each node will choose one of its surrounding tags to update itself [24]. Although the time complexity of this method of selecting labels through many iterations is low, the randomness is high. After the termination condition is reached, nodes holding the same label are divided into the same community.

Louvain. Louvain divides every node into separate communities, and each node is moved between communities to achieve the purpose of community division. The algorithm will be stopped when moving any one node can improve overall modularity degree scores [33].

Leading eigenvector. LE is a community detection algorithm using the leading eigenvector method, which divides the network structure by continuously maximizing the original network modularity degree [34].

FluidC. FluidC is probably the first community detection algorithm based on the idea of fluid dyeing [35]. This algorithm needs to determine the number of communities in advance.

EDCD. EDCD is a new algorithm for iteratively deleting constrained edges [36]. The original network is divided into several vertices of strongly connected communities and the community structure is optimized by an improved edge deletion process optimization module. Finally, the isolated vertex initial communities are reconnected to optimize the community structure.

Table 2. The basic principle and time complexity of algorithms used for comparison.

Algorithm	Basic Principle	Time Complexity
FG	Modularity optimization	$O(e(e+n))$
SG	Spin configuration	$O((k+e)n)$
Infomap	Random walk	$O(e)$
LPA	Label propagation	$O(n)$
Louvain	Separate division	$O(kn)$
LE	Eigenvectors of matrices	$O(n^2)$
FluidC	Fluids interacting in an environment	$O(e)$
EDCD	Edge deletion	$O(c(k+1)e+kn)$
CDPD	Preferential decision	$O(t \times k^2 \times n)$

4.2. Data Description

The datasets used in the experiments include synthetic network datasets and real-world network datasets. Synthetic network datasets generated by an LFR model represent the virtual network in the network. Real-world network datasets come from real life.

4.2.1. Synthetic Networks

We use the well-known LFR model [37], which can fit the real-world network characteristics by controlling several parameters of the LFR model, such as number of nodes ($|V|$), average degree (k), maximum degree ($maxk$), minimum community size ($minc$), and maximum community size ($maxc$). Among these, what is particularly noteworthy here is the mixing parameter (mu), which indicates whether the community structure in the network is obvious. It can be specifically defined as:

$$mu = \frac{k_{out}}{k}, \tag{7}$$

where k represents the number of connections for all nodes and k_{out} represents the number of connections between nodes in different communities. The higher the value of mu is, the more blurred the community structure of the network is. We want to observe the performance of several algorithms on different mu. We use the LFR model to generate a series of networks where the number of nodes $|V| = 1000$, the average degree $k = 15$, the maximum degree $maxk = 38$, the minimum community size $minc = 10$, and the maximum community size $maxc = 50$. We set mu increase from 0.1 to 0.8. At the same time, the average degree k may also be an important factor influencing the results of community detection. We set $mu = 0.1$ and the average degree k changes from 3 to 20 to generate several networks. Based on this experimental idea, two groups of synthetic networks have been generated as Table 3:

Table 3. Characteristics of synthetic networks.

Dataset	$\|V\|$	k	$maxk$	mu	$minc$	$maxc$
1st	1000	15	38	0.1–0.8	10	50
2nd	1000	3–20	$6 * k$	0.1	10	50

4.2.2. Real-World Networks

We also select some representative real networks to evaluate the above algorithms. All data of networks can be downloaded from SNAP (http://snap.stanford.edu/data/) and UCI (https://networkdata.ics.uci.edu/index.php). Some relevant information about real-world networks have be briefly described in the following Table 4.

Table 4. Characteristics of real-world networks.

Datasets	$\mid V \mid$	$\mid E \mid$	k	$\mid C \mid$	Clustering Coefficient	Average Path Length
Karate	34	78	4.588	2	0.588	2.408
Dolphins	62	159	5.129	2	0.303	3.357
Football	115	613	10.611	12	0.403	2.508
Polbooks	105	441	8.4	3	0.48	3.097
Citeseer	2114	7396	3.52	6	0.24	9.32

Zacharys Karate network. The Karate network is a social network constructed by observing an American University Karate club. The network consists of 34 nodes and 78 edges, in which the node represents the member of the club and the edge represents the friendship between the members.

Dolphin social network. A group of researchers from the Harbor Branch Oceanographic Institution ($HBOI$) carefully studied the relationship between Dolphins in the Indian lagoon. The network consists of 62 nodes and 159 edges. To better understand how dolphin populations treat and use their environment, scientists can determine how social networks affect information transmission and potentially affect reproductive behavior and disease transmission.

Politics books network. This is a book network of American politics that are sold by the online bookseller on Amazon.com, consisting of 105 nodes and 441 edges. The nodes represent US politics-related books sold on the online bookstore and the edges represent some readers who purchased both books at the same time.

Football network. This network is a real network based on the American College Football League. The network consists of 115 nodes and 616 edges. The node represents the team and the edge represents the game between the two teams. These teams are divided into 12 leagues, which can be analogous to our real online community. Similarly, this network is a classic datasets in community detection.

Citeseer network. This network is derived from a scientific paper citation dataset. The node represents the paper and the edge represents the mutual reference relationship between the two papers. The network is mainly divided into six categories, which contains 2110 nodes and 4732 edges.

4.3. Evaluation Metrics

Evaluating the result of community detection has raised a fierce discussion in the data mining area, and some of the classic evaluation indicators have been put forward in recent years, which provides a reference for our research. However, some scholars have confirmed that two different evaluation indicators may have the opposite conclusions on the same experimental results [8]. This experiment selects multiple commonly used evaluation indexes to evaluate the above nine algorithms community detection areas, such as Normalized Mutual Information (NMI) [38], Adjusted Rand Index (ARI) [39] and Cluster Purity ($purity$) [40].

Normalized Mutual Information. NMI is a formula that represents the similarity of two sets, which stems from the information theory. The specific definition is as follows:

$$NMI(X,Y) = \frac{-2\sum_{i=1}^{C_X}\sum_{j=1}^{C_Y} N_{ij} log(\frac{N_{ij}N}{N_i N_j})}{\sum_{i=1}^{C_X} N_i log(\frac{N_i}{N}) + \sum_{j=1}^{C_Y} N_j log(\frac{N_j}{N})}, \tag{8}$$

where X represents the real partition and Y is the partition found by the algorithm; C_X is the number of real communities and C_Y is the number of found communities; N is the number of nodes in the network; N_{ij} is the number of nodes shared by the real community i in partition X and the found community j in partition Y; N_i denotes the sum over row i of matrix N_{ij}; and N_j denotes the sum over column j [38].

Adjusted Rand Index. ARI is another commonly used clustering evaluation index based on similarity measure, which is defined as:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}] / \binom{n}{2}}. \tag{9}$$

if you want to know more details, please refer to [41,42].

Cluster Purity. $Purity$ is a evaluation indicator that can measure the accuracy of classification, which is defined as:

$$Purity(\Omega, C) = \frac{1}{N}\sum_{i=1}^{k} max_j |\omega_i \cap c_j|, \tag{10}$$

where N is the number of community, $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$ represents the set of predicted community, and $C = \{c_1, c_2, \ldots, c_j\}$ is the set of ground truth.

4.4. Performance Evaluation

In all experiments, the parameters of the comparison algorithm are set as the default parameters suggested by the author, putting into mind if the parameters are even needed. For a dataset with known community partition results, we enter the number of communities as a parameter for FluidC. For a dataset with an unknown number of communities, we use the number of communities found by the CDPD as a parameter input. Most of the comparison algorithms are based on the igraph package (http://igraph.org/) except for FluidC which can be downloaded on Github (github.com/FerranPares/Fluid-Communities).

Evaluation of Synthetic Networks. We apply the above algorithms to the LFR synthetic network datasets, where the mu of the datasets varies from 0.1 to 0.8. We evaluate the performance of the algorithm by using various aspects of the evaluation indicators (NMI, ARI, $Purity$) and the recorded experiment results have been shown in Figure 2.

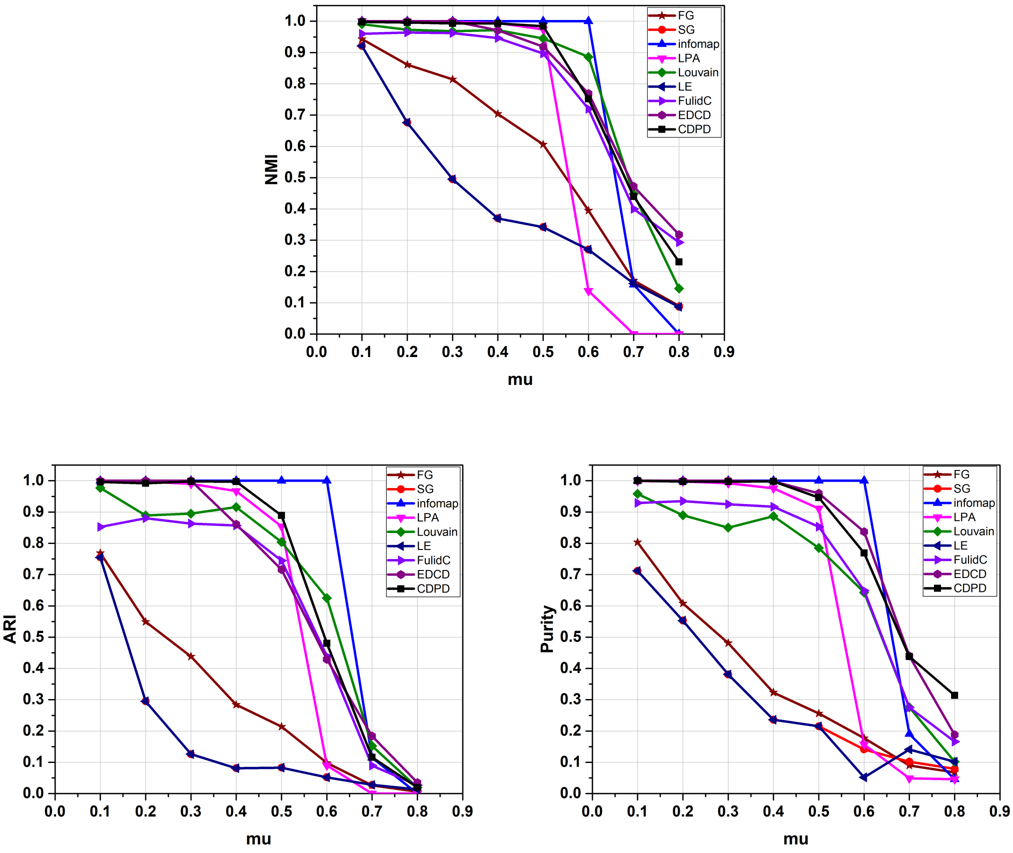

Figure 2. Performances of the above algorithms on the synthetic networks generated by LFR by varying the mixing parameter from mu 0.1 to 0.8.

We can see that all three evaluation indicators decrease with increasing mu because the growth of mu means that the community structure is increasingly blurred. In Figure 2, when mu is less than or equal to 0.4 $(mu \leq 0.4)$, CDPD always maintains optimal performance when compared with other algorithms. When mu is equal to 0.5 $(mu = 0.5)$, the performance of CDPD is second only to Infomap. We can see that Infomap always performs well with mu ranging from 0.1 to 0.6. However, when mu equals 0.7 $(u = 0.7)$, the performance of Infomap decays rapidly; although the performance of CDPD has also weakened, it is still relatively superior.

Then, to verify the performance of above algorithms to network on the different average degree k, we run these algorithms on the LFR synthetic networks when k is varied from 3 to 20. The results obtained are shown in Figure 3. From Figure 3, we can see that the performance of LE is always poor compared to other algorithms, but other algorithms perform better with increasing k. When k is greater than or equal to 8 $(k \geq 8)$, CDPD has the best results as other algorithms. When k is equal to 5 $(k = 5)$, the performance of CDPD on NMI and $Purity$ is still optimal, and the performance of ARI is second only to EDCD and LPA. When k is equal to 3 $(k = 3)$, CDPD is superior to other algorithms on all indicators and has obvious advantages.

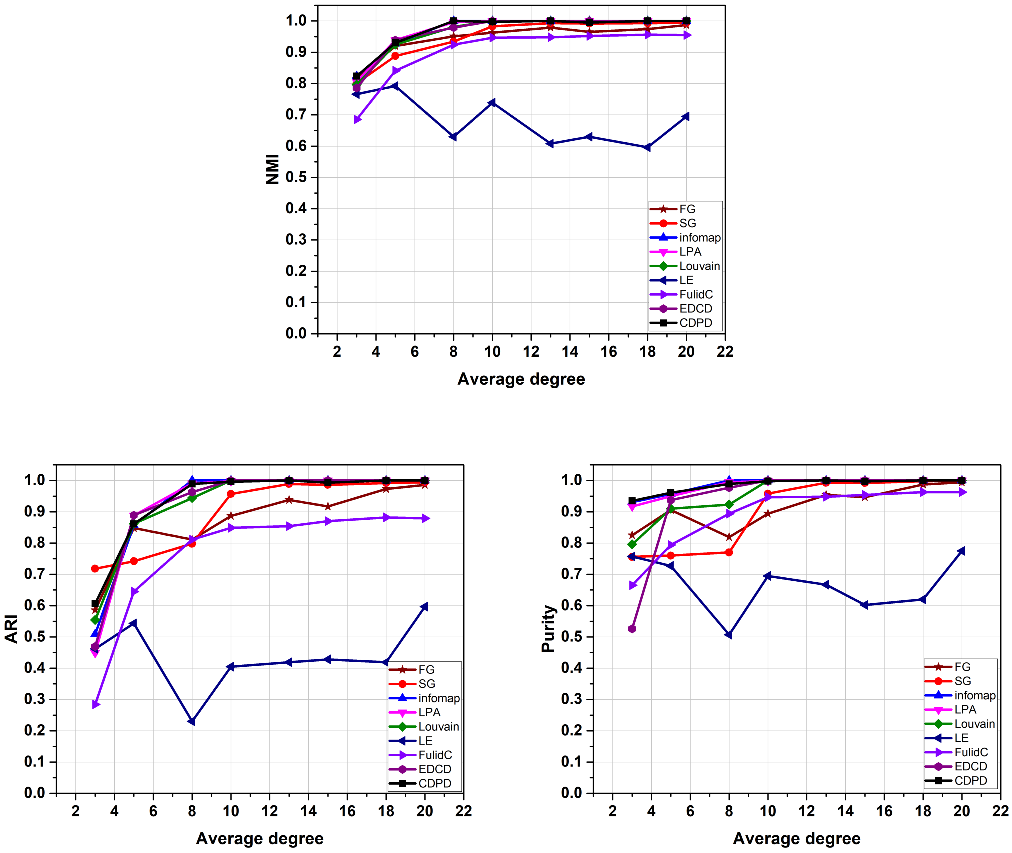

Figure 3. Performances of the above algorithms on the synthetic networks generated by LFR by varying the average degree k from 3 to 20.

Evaluation of Real-World Networks. To more widely verify the performance of the above algorithms, we apply these algorithms to real-world networks with ground truth. The results obtained are shown in Table 5. The best performance on each of the datasets is bolded for observation. In Karate, CDPD has a good performance on all metrics $(NMI = 0.897, ARI = 0.902, Purity = 1)$, which is the best among all algorithms. Infomap has performed well in synthetic network datasets but performs poorly here. The detection results are visualized as shown in Figure 4a. We can see that the neighbor nodes of node 10, node 3, and 34, belonging to two communities respectively. According to the label update rules of LPA, when the neighbor node has multiple maximum numbers of labels, one of the nodes will be randomly selected by node 10. Node 10 in the LPA has a high probability of being divided into the wrong community. The CDPD prefers to learn the labels of neighbors with a higher degree. This ensures that node 10 and node 34 has a greater probability is divided into the same community because node 34 has a greater degree. In Dolphins, CDPD has the best NMI and ARI $(NMI = 0.72, ARI = 0.69)$. However, the $Purity$ of CDPD $(Purity = 0.984)$ is inferior to $Purity$ of SG $(Purity = 0.999)$. However, the NMI $(NMI = 0.638)$ and ARI $(ARI = 0.493)$ of SG are smaller than CDPD. Therefore, the overall performance of CDPD in Dolphins is good. In addition, the community detection results of Dolphins are shown in Figure 4b. In Polbooks, the performance of CDPD is still outstanding. The NMI of CDPD $(NMI = 0.586)$, ARI $(ARI = 0.69)$ and $Purity$ $(Purity = 0.924)$ are all the best. In addition, the community division results of Polbooks are shown in Figure 4c. In Football, CDPD, EDCD, and Infomap have the best NMI $(NMI = 0.92)$, and CDPD and Infomap also have the best $Purity$ $(Purity = 0.92)$. However, the ARI of CDPD $(ARI = 0.85)$ is lower than ARI of Infomap $(ARI = 0.897)$. In addition, the community division results of Football are shown in Figure 4d. Finally, in Citeseer, the ARI of CDPD and FG $(ARI = 0.198)$ is the best. However, the

NMI of the CDPD $(NMI = 0.398)$ is only lower than the NMI of the EDCD $(NMI = 0.435)$, which performs better than other algorithms. The $Purity$ of the CDPD $(Purity = 0.492)$ is slightly lower than the $Purity$ of Infomap $(Purity = 0.777)$.

In summary, compared to other comparison algorithms, CDPD has a good performance in both the synthetic networks and the real-world networks. Especially in networks with real-world networks and super sparse networks, CDPD can still achieve good partitioning results when other algorithms struggle (cf. Figure 4 and Table 5).

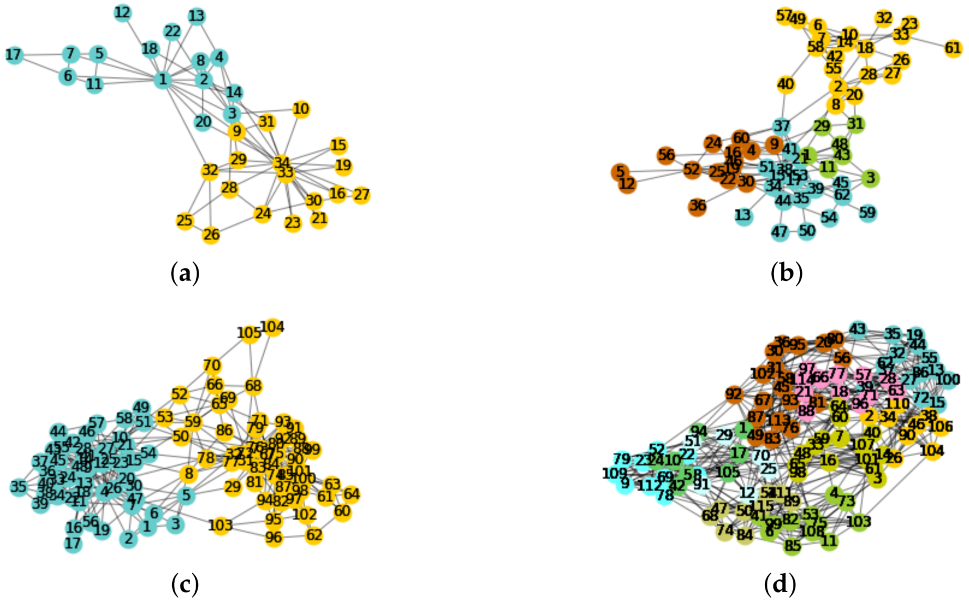

Figure 4. The communities obtained in different networks by CDPD. (**a**) community structure obtained by CDPD on Karate, (**b**) community structure obtained by CDPD on Dolphins, (**c**) community structure obtained by CDPD on Polbooks, (**d**) community structure obtained by CDPD on Football.

Table 5. Performances of the above algorithms in real-world networks with ground truth.

	Zachary			Dolphins			Polbooks			Football			Citeseer		
	NMI	**ARI**	**Pur.**	**NMI**	**ARI**	**Pur.**	**NMI**	**ARI**	**Pur.**	**NMI**	**ARI**	**Pur.**	**NMI**	**ARI**	**Pur.**
FG	0.707	0.68	0.971	0.652	0.493	0.984	0.531	0.638	0.838	0.757	0.474	0.583	0.362	**0.198**	0.705
SG	0.703	0.526	0.994	0.638	0.369	**0.999**	0.511	0.51	0.851	0.906	0.846	0.897	0.28	0.16	0.62
Infomap	0.711	0.702	0.971	0.611	0.352	0.991	0.503	0.536	0.848	**0.92**	**0.897**	**0.922**	0.397	0.033	**0.777**
LPA	0.664	0.642	0.913	0.679	0.556	0.978	0.556	0.652	0.845	0.878	0.756	0.846	0.388	0.071	0.776
Louvain	0.618	0.462	0.971	0.568	0.343	0.968	0.513	0.549	0.848	0.891	0.807	0.87	0.352	0.159	0.698
LE	0.715	0.512	**1**	0.497	0.283	0.952	0.525	0.547	0.848	0.703	0.464	0.626	0.341	0.176	0.666
FluidC	0.642	0.594	0.76	0.631	0.628	0.886	0.496	0.532	0.751	0.885	0.79	0.866	0.239	0.197	0.482
EDCD	0.896	0.88	0.968	0.37	0.13	0.387	0.52	0.56	0.790	**0.92**	0.896	0.921	**0.435**	0.13	0.243
CDPD	**0.897**	**0.902**	**1**	**0.72**	**0.69**	0.984	**0.586**	**0.69**	**0.924**	**0.92**	0.85	**0.922**	0.398	**0.198**	0.492

5. Conclusions

In this paper, we introduced a community detection algorithm, called CDPD, which is based on the preferential decision model. CDPD does not need any parameters to uncover community structure, and it has respectable accuracy in most cases. It especially performs well in small networks and sparse networks. At the same time, another advantage of CDPD is that it is stable. To put it in a nutshell, CDPD has the advantage of higher accuracy, free parameter, and stability. In the future, we intend to apply our algorithm to overlapping community detection and detection based on dynamic networks and large-scale network applications. Further improving the efficiency of the algorithm is still a challenging and meaningful topic.

Author Contributions: Conceptualization, J.S. and B.W.; Software, B.L., J.H. and Q.D.; Supervision, J.S. and B.W.; Validation, B.L., J.H., K.W. and D.A.; Writing, J.S., B.L. and X.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Major Project of China (2017ZX06002005).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oger, R.; Yerson, B.M. Game theory: Analysis of conflict. *Long Range Plan.* **1992**, *25*, 130.
2. Basu, S.; Maulik, U. Community detection based on strong Nash stable graph partition. *Soc. Netw. Anal. Min.* **2015**, *5*, 1–15. [CrossRef]
3. Fortunato, S.; Barthelemy, M. Resolution limit in community detection. *Mob. Netw. Appl.* **2007**, *104*, 36–41. [CrossRef] [PubMed]
4. *IEEE Transactions on Emerging Topics in Computational Intelligence*; Publishing House: New York, NY, USA, 2018; Volume 2, pp. 214–223.
5. Li, H.J.; Daniels, J.J. Social significance of community structure: Statistical view. *Phys. Rev. E* **2015**, *91*, 012801. [CrossRef] [PubMed]
6. Bu, Z.; Li, H.J.; Zhang, C.; Cao, J.; Li, A.; Shi, Y. Graph K-means based on Leader Identification, Dynamic Game and Opinion Dynamics. *IEEE Trans. Knowl. Data Eng.* **2019**, doi:10.1109/TKDE.2019.2903712. [CrossRef]
7. Li, H.J.; Bu, Z.; Wang, Z.; Cao, J. Dynamical clustering in electronic commerce systems via optimization and leadership expansion. *IEEE Trans. Ind. Inform.* **2019**, doi:10.1109/TII.2019.2960835. [CrossRef]
8. Fortunato, S.; Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1–44. [CrossRef]
9. Khan, B.S.; Niazi, M.A. Network Community Detection: A Review and Visual Survey. *arXiv* **2017**, arXiv:1708.00977.
10. Jonnalagadda, A.; Kuppusamy, L. A survey on game theoretic models for community detection in social networks. *Soc. Netw. Anal. Min.* **2016**, *6*, 83. [CrossRef]
11. Qi, X.; Tang, W.; Wu, Y.; Guo, G.; Fuller, E.; Zhang, C.Q. Optimal local community detection in social networks based on density drop of subgraphs. *Pattern Recognit. Lett.* **2014**, *36*, 46–53. [CrossRef]
12. Badie, R.; Aleahmad, A.; Asadpour, M.; Rahgozar, M. An efficient agent-based algorithm for overlapping community detection using nodes closeness. *Phys. Stat. Mech. Its Appl.* **2013**, *392*, 129–140. [CrossRef]
13. Newman, M.E.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2004**, *69*, 026113. [CrossRef] [PubMed]
14. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [CrossRef]
15. Guimera, R.; SalesPardo, M.; Amaral, L.A. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2004**, *70*, 025101. [CrossRef]
16. Jordi, D.; Alex, A. Community Detection in Complex Networks Using Extremal Optimization. *Phys. Rev. Stat. Nonlin. Soft. Matter. Phys.* **2005**, *72*, 027104.
17. Osborne, M.J.; Rubinstein, A. *A Course in Game Theory*; MIT Press: Cambridge, MA, USA, 1994.
18. Li, H-.J.; Wang, Q.; Liu, S.; Hu, J. Exploring the trust management mechanism in self-organizing complex network based on game theory. *Phys. Stat. Mech. Its Appl.* **2019**, 123514. [CrossRef]
19. Cao, J.; Bu, Z.; Wang, Y.; Yang, H.; Jiang, J.; Li, H.-J. Detecting Prosumer-Community Groups in Smart Grids From the Multiagent Perspective. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, 1–13. [CrossRef]
20. Narayanam, R.; Narahari, Y. A game theory inspired, decentralized, local information based algorithm for community detection in social graphs. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 1072–1075.
21. Cao, L.; Li, X.; Han, L. Detecting community structure of networks using evolutionary coordination games. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS2013), Beijing, China, 19–23 May 2013; pp. 2533–2536.

22. Alvari, H.; Hajibagheri, A.; Sukthankar, G. Community detection in dynamic social networks: A game-theoretic approach. In Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing China, 17–20 August 2014; pp. 101–107.
23. Pons, P.; Latapy, M. Computing Communities in Large Networks Using Random Walks. *J. Graph Algorithms Appl.* **2006**, *10*, 191–218. [CrossRef]
24. Raghavan, U.N.; Albert, R.; Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2007**, *76*, 036106. [CrossRef]
25. Hosseini, R.; Azmi, R. Memory-based label propagation algorithm for community detection in social networks. In Proceedings of the International Symposium on Artificial Intelligence and Signal Processing, Mashhad, Iran, 3–5 March 2015; pp. 256–260.
26. Cordasco, G.; Gargano, L. Community Detection via Semi-Synchronous Label Propagation Algorithms. In Proceedings of the 2010 IEEE International Workshop on: Business Applications of Social Network Analysis (BASNA), Bangalore, India, 5 December 2010; pp. 1–8.
27. Zhang, X.; Kun, R.; Jing, S.; Chen, J.; Zhang, Q. Label propagation algorithm for community detection based on node importance and label influence. *Mob. Phys. Lett.* **2017**, *381*, 2691–2698. [CrossRef]
28. Khadivi, A.; Rad, A.A.; Hasler, M. Community detection enhancement in networks using proper weighting and partial synchronization. In Proceedings of the IEEE International Symposium on Circuits and System, Florence, Italy, 27–30 May 2018; pp. 3777–3780.
29. Shao, J.; Han, Z.; Yang, Q.; Zhou, T. Community Detection based on Distance Dynamics. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 8–9 August 2015; pp. 1075–1084.
30. Clauset, A.; Newman, M.E.; Moore, C. Finding community structure in very large networks. *Phys. Rev. Stat. Nonlinear Soft. Matter. Phys.* **2004**, *70*, 066111. [CrossRef] [PubMed]
31. Reichardt, J.; Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2006**, *74*, 016110. [CrossRef] [PubMed]
32. Rosvall, M.; Bergstrom, C.T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 1118–1123. [CrossRef] [PubMed]
33. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of community hierarchies in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, 10008. [CrossRef]
34. Newman, M.E.J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2006**, *74*, 036104. [CrossRef]
35. Parés, F.; Gasulla, D.G.; Vilalta, A.; Moreno, J.; Ayguadé, E.; Labarta, J.; Cortés, U.; Suzumura, T. Fluid Communities: A Competitive, Scalable and Diverse Community Detection Algorithm. In Proceedings of the Complex Networks and Their Applications VI, Lyon, France, 29 November–1 December 2017; pp. 229–240.
36. Chen, X.; Li, J. Community detection in complex networks using edge-deleting with restrictions. *Phys. Stat. Mech. Its Appl.* **2019**, *519*, 181–194. [CrossRef]
37. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. Stat. Nonlinear Soft Matter Phys.* **2008**, *78*, 046110. [CrossRef]
38. Strehl, A.; Ghosh, J. Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583-617.
39. Rand, W.M. Objective Criteria for the Evaluation of Clustering Methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850. [CrossRef]
40. Ying Z.; Karypis, G. Criterion Functions for Document Clustering: Experiments and Analysis. 2002. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.6872 (accessed on 16 January 2020).
41. Vinh, N.X.; Epps, J.; Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* **2010**, *11*, 2837–2854.
42. Lawrence, H.; Phipp, A. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.