Article

New Random Walk Algorithm Based on Different Seed Nodes for Community Detection

Wencong Li , Jiansheng Cai * , Xiaodong Zhang , Jihui Wang

Posted Date: 8 July 2024

doi: 10.20944/preprints202407.0562.v1

Keywords: Complex networks; Community detection; Random walk; Seed nodes

Article

New Random Walk Algorithm Based on Different Seed Nodes for Community Detection

Wencong Li [1,†], Jiansheng Cai [2,*,†], Xiaodong Zhang [3] and Jihui Wang [4]

[1] School of Mathematical Sciences, Ocean University of China, Songling Road, Qingdao, 266100, China, li_ujn@126.com
[2] School of Mathematics and Statistics; Weifang University, Dongfeng East Street, Weifang, 261061, China, healthcai@163.com
[3] School of Mathematical Sciences, MOE-LSC, SHL-MAC, Shanghai Jiao Tong University, Dongchuan Road, Shanghai, 200240, China, xiaodong@sjtu.edu.cn
[4] School of Mathematical Sciences, University of Jinan, West Nanxinzhuang Road, Jinan, 250022, China, ss_wangjh@ujn.edu.cn
[*] Correspondence: healthcai@163.com
[†] These authors contributed equally to this work.

Abstract: The complex network is an abstract modeing of complex systems in the real world, which plays an important role in analyzing the function of complex systems. Community detection is an important tool for analyzing network structure. In this paper, we propose a new community detection algorithm (RWBS) based on different seed nodes aims to understand the community structure of the network, which provides a new idea for the allocation of resources in the network. RWBS provides a new centrality metric (MC) to calculate node importance, which calculates the ranking of nodes as seed nodes. Furthermore, two algorithms are proposed for determining seed nodes on networks with and without ground-truth, respectively. We set the number of steps for the random walk to 6 according to the six degrees of separation theory to reduce the running time of the algorithm. Since some traditional community detection algorithms may detect smaller communities, e.g., two nodes become one community, this may make the resource allocation unreasonable. Therefore, modularity (Q) is chosen as the optimization function to combine communities, which can improve the quality of detected communities. Final experimental results on real-world and synthetic networks show that the RWBS algorithm can effectively detect communities.

Keywords: complex networks; community detection; random walk; seed nodes

1. Introduction

In recent years, complex networks have been widely studied because of its important applications in reality. It is a special mathematical model that considers the relationships between objects in the real world. Moreover, it generally has three properties: (1) The small world property [1], which describes the shortest distance between any two nodes on the network is short. (2) Scale-free property [2,3], where the degree distribution of nodes conforms to a power rate distribution, i.e., most of the nodes have smaller degree and very few nodes have larger degree. (3) Community structure [4,5], i.e., the nodes on the network exhibit the characteristics of clusters. It is a subgraph structure with tight internal connections and sparse external connections. Most networks are characterized with community structure. For example, communities in the social network [6,7] represent closely related groups. Communities in the citation network [8] represent clusters of articles in a particular field of study. Communities in the protein-to-protein interaction network [9] represent clusters of proteins with similar biological functions. Therefore, community detection is gradually becoming an important research area in complex networks.

Some community detection algorithms based on random walk [10,11] need to set a convergence condition and end the random walk when this condition is satisfied, which may spend a longer time. The label propagation algorithm (LPA) [12] has a disadvantage in detecting communities, i.e., the randomness of label propagation. It may detect poor quality communities. The proposed algorithm uses random walk and the label propagation while addressing its disadvantages.

In this paper, a random walk algorithm based on different seed nodes to detect communities is proposed, named RWBS. A new metric is proposed to measure the importance of nodes, we can get the similarity $F(i,j)$ between any two nodes on the network by this metric. Based on $F(i,j)$, we propose two algorithms that are suitable for different types of networks to obtain seed nodes of the random walk. Moreover, RWBS changes the transfer probability matrix to get better information about the network. According to the six degrees of separation theory in complex networks, the steps of random walk is set to 6, which not only can shorten the time of the random walk, but also can obtain the structural information of the network. After calculating the weight of the edge, the new label propagation rule is used to propagate the label, and the initial community structure is obtained based on the convergent label. Finally, the modularity (Q) is chosen as the optimization function to further combine communities and improve communities quality. Experiments on real-world and synthetic networks demonstrate the effectiveness and superiority of RWBS. The following are the main innovations of paper.

- We propose a random walk algorithm based on different seed nodes to detect communities, named RWBS.
- We propose a new centrality metric (MC) for measuring the importance of nodes that combines degree centrality ($DC^{'}$) and closeness centrality (CC) and performs better than ($DC^{'}$) and (CC).
- Two algorithms are proposed to obtain seed nodes for different networks.
- Experimental results on real-world and synthetic networks show that the RWBS algorithm can be effective in finding communities.

2. Related Works

With the intensive study of community structure in complex networks, more and more effective community detection algorithms have been proposed. Below, we describe each of these methods.

2.1. The Traditional Methods

The Traditional methods can be meticulously classified into the following three methods.

- The partitional clustering method: The method is to divide the network into K subgraphs of predefined size such that the edges within each subgraph are denser and the edges between different subgraphs are sparser. Commonly used algorithms are KL algorithm [13] and spectral bisection algorithm [14]. The disadvantage of this class of methods is that the size of the community needs to be set in advance. However, real-world networks are largely unknown about their community structure, making it difficult to apply it in practice.
- The hierarchical clustering methods: Networks can be represented by adjacency matrices or small matrices after dimensionality reduction such as matrix factorization transformation, and then clustered using conventional clustering algorithms. The first method is the hierarchical clustering approach, which considers a graph to be a large community that contains a complex topology, i.e., the community may be a collection of smaller communities of different sizes [15]. Another method is that of spectral clustering, which consists of the method of using matrix eigenvectors and the method of classifying nodes based on pairwise similarities between data points [16]. In 2022, Ullah et al. proposed an Information Interaction Model, named RIIM algorithm [17].
- The divisive method: This method obtains the community structure by calculating the similarity of edges to remove edges with lower similarity [18]. The entire network is first categorized into a community, and then the edges connecting the low-similarity vertices and the highest edge interdimensionality are removed. The method is a top-down hierarchical clustering algorithm. For example, the time complexity of the GN algorithm [5] proposed by Girvan and Newman in 2002 is $O(n^3)$. The disadvantage of this algorithm is the high time complexity, which makes it difficult to be applied to large networks.

2.2. The Modularity-Based Methods

Modularity [19] can be used to measure the quality of the community. A higher value of modularity also means that the quality of the community obtained is better. Many scholars have used the modularity as the optimization function to achieve the optimal division result when dividing communities.

- Greedy optimization: In 2004, Newman [20] proposed a greedy method for maximizing modularity, which was an aggregation technique. Its time complexity on the sparse graph is $O((m+n)n)$. Another greedy optimization algorithm is an algorithm called Louvain (BGLL) [21] proposed by Blondel et al. Its time complexity is $O(m+n)$.
- Simulated annealing: It is a globally optimized discrete stochastic method to detect communities in complex networks by maximizing the modularity. For example, Guimerà and Amaral [22] proposed an annealing modularity optimization algorithm (SA) based on the principle of simulated annealing algorithm.
- Extremal optimization: In 2001, Boettcher [23] et al. proposed extremal optimization as a general heuristic search technique for physical and combinatorial optimization problems. In 2005, Duch et al. [24] used it for modularity optimization to detect communities.
- Genetic algorithms: Genetic algorithms are optimization techniques inspired by biological evolution. They can also be used to optimize the modularity to detect communities. For example, in 2018, M'Barek [25] et al. proposed a Genetic Algorithm (GA) based approach to find communities in a gene interaction network. In 2019, a new matrix-based genetic algorithm for community detection was proposed by Chen and Bi, named MGA algorithm [26].
- Evolutionary algorithms: It is a type of metaheuristic optimization algorithm based on artificial intelligence. Their effectiveness in local learning and global search is well known. For example, in 2021, Pourabbasi et al. proposed a single-chromosome evolutionary algorithm combining content and structural information to detect communities [27]. Su et al. proposed a parallel multi-objective evolutionary algorithm, called PMOEA [28].

2.3. The Dynamic Community Detection Methods

We introduce three dynamic community detection methods.

- Algorithms based on random walk: In the process of random walk, the random walker starts walking within the community from one node and randomly moves to the neighboring nodes at each step. The random walker spends a long time in the dense community because of the dense edge connections within the community. The algorithms based on random walk are PageRank algorithm [29], Walktrap [10], and Infomap [30].
- Algorithms based on LPA [12]: The LPA algorithm is widely used to find communities in large networks due to its advantages of lower time complexity and space complexity. The detailed steps of LPA are shown in Section 3.4. Since the LPA algorithm has the advantage of low time complexity, many scholars have researched and proposed many LPA-based community division algorithms based on this algorithm. For example, SLPA [31], LPA_CL [32], VLPA [33], etc..
- Other algorithms: There are a few other dynamic community division algorithms. For example, the CDME algorithm that is based on the Matthew effect [34], The GBTM algorithm for community detection in dynamic networks by Hidden Markov Method [35].

3. Preparation of Algorithm

3.1. The Definition of Symbols

The community division algorithm proposed in this paper is proposed for undirected and unweighted networks. Therefore, all the datasets used in this paper are undirected and unweighted networks. We assume that the network is G. The set of nodes and edges of the network can be defined as $V(G)$ and $E(G)$, respectively. Then, G can be expressed by $V(G)$ and $E(G)$, i.e., $G = (V(G), E(G))$. The definitions of other mathematical symbols used in this paper are shown in Table 1.

Table 1. The definitions of other mathematical symbols used in this paper.

Symbol	Definition
n	The number of nodes
m	The number of edges
$k(i)$	The degree of node i
l_i	The label of node i
$community(i)$	The community to which node i belongs
c	The number of communities.
$N(i)$	The neighbor nodes of node i
α	The restart probability
$L(\lambda)$	The set of labels that have been updated λ times
k_{max}	The maximum degree of the nodes on the network
$< k >$	The average degree of the network

3.2. Evaluation Metrics

We use Normalized Mutual Information (NMI) [36], Adjusted Rand Index (ARI), and Modularity (Q) [19] as the evaluation metrics to measure the quality of communities. Let $X = (C_1^{'}, C_2^{'}, ..., C_p^{'})$ and $Y = (C_1, C_2, ..., C_c)$ represent the detected community by the proposed algorithm and the real community of the network, respectively. All three metrics are such that a larger value represents a better quality community obtained. The specific descriptions of these metrics are shown below.

- NMI [36]: Where NMI is used to measure the similarity of the communities detected by our proposed algorithm with the real communities of the network. The definition of NMI is shown below.

$$NMI(X, Y) = \frac{2I(X;Y)}{H(X) + H(Y)} \tag{1}$$

  $H(X)$ and $H(Y)$ are the entropies of X and Y. $I(X;Y)$ represents the mutual information between X and Y.
- ARI [37]: Similarly, ARI is used to measure the similarity of the communities detected by our proposed algorithm with the real communities of the network. It is defined as Eq. (2).

$$ARI = \frac{\sum\limits_{i,j}\binom{n_{ij}}{2} - [\sum\limits_{i}\binom{a_i}{2}\sum\limits_{j}\binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum\limits_{i}\binom{a_i}{2} + \sum\limits_{j}\binom{b_j}{2}] - [\sum\limits_{i}\binom{a_i}{2}\sum\limits_{j}\binom{b_j}{2}]/\binom{n}{2}} \tag{2}$$

  Where $n_{ij} = |C_i^{'} \cap C_j|$, $a_i = |C_i^{'}|$, and $b_j = |C_j|$ ($i \in \{1, 2, ..., p\}$, $j \in \{1, 2, ..., c\}$).
- Q [19,38]: The modularity is defined as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k(i)k(j)}{2m}\right]\delta_{c_i,c_j} \tag{3}$$

  Where c_i represents the community to which node i belongs. The network G can be described by the adjacency matrix $A = (A_{ij})_{n\times n}$, $A_{ij} = 1$ if $(i,j) \in E(G)$ and 0 otherwise. δ_{c_i,c_j} is 1 if $c_i = c_j$ and 0 otherwise.

3.3. The Importance of Nodes

Measuring the importance of nodes is also an important issue in the field of complex networks and has a wide variety of applications. Some centralities are important metrics for measuring the importance of nodes. For example, degree centrality [39], betweenness centrality [40], closeness centrality [41], eigenvector centrality [42], pagerank centrality [43], etc. The centralities used in this paper are described below.

3.3.1. Degree Centrality

Degree centrality [39] measures the importance of a node by its degree. Higher degree of a node means that the node is more important. It is defined as Eq. (4).

$$DC(i) = \frac{k(i)}{n} \tag{4}$$

However, the above definition does not take into account the size of the network. To solve this problem, Stanley Wasserman and Katherine Faust [44] proposed the standardized degree centrality. It can be described as:

$$DC^{'}(i) = \frac{k(i)}{n-1} \tag{5}$$

3.3.2. Closeness Centrality

This centrality [41] can reflect the closeness between two nodes. It is defined as follows:

$$CC(i) = \frac{1}{\sum\limits_{j \in V(G), j \neq i} d(i,j)} \tag{6}$$

where $d(i,j)$ represents the shortest distance between node i and node j. When the sum of the shortest distances from node i to other nodes on the network is shorter, then its closeness centrality is higher.

3.3.3. Mixed Centrality

To fully combine both of these centrality, we introduce the mixed centrality. $MC(i)$ is defined as Eq. (7).

$$\begin{aligned} MC(i) &= \frac{e^{DC^{'}(i)}}{max\left\{e^{DC^{'}(i)} : i \in V(G)\right\}} * \frac{e^{CC(i)}}{max\{e^{CC(i)} : i \in V(G)\}} \\ &= \frac{e^{[DC^{'}(i)+CC(i)]}}{max\left\{e^{[DC^{'}(i)+CC(j)]} : i,j \in V(G)\right\}} \end{aligned} \tag{7}$$

We test the effectiveness of the mixed centrality. Network efficiency reflects how well-connected the network is. As a general rule, the better the network connection, the more efficient the network [45]. The following is its definition.

$$\eta = \frac{1}{n(n-1)} \sum_{i \neq j \in V(G)} \frac{1}{d(i,j)} \tag{8}$$

Then, we observe the decline rate of network efficiency by deleting node i as a way to determine whether the node's importance ranking in the network is justified. Let λ_i be the decline rate of network efficiency:

$$\lambda_i = 1 - \frac{\eta_i}{\eta_0} \tag{9}$$

where η_i is the efficiency of the network after removing node i and η_0 is the original efficiency of the network. The larger λ_i is, the more important the removed node i is. It should occur with a clear correlation between the decline rate of network efficiency and the importance of a node as the importance of a node decreases. As an explanation of the decline rate of network efficiency when the node is removed, Figure 1 is shown. Figure 2 shows the correlation between the decline rate of network efficiency and the importance of nodes ranked by $DC^{'}$, CC, and MC. From Figure 2(c), it can be seen that overall the rate of decrease in network efficiency decreases as the MC value of the deleted node decreases. It is clear that the decline rate of the network efficiency correlates with the importance of the node for $DC^{'}$ and MC methods, and MC performs better. The performance of CC is the worst, and in Figure 2(b), the decline rate of the network efficiency and the importance of the node do not appear to be correlated. Therefore, we propose MC as an effective metric to measure the importance of nodes.

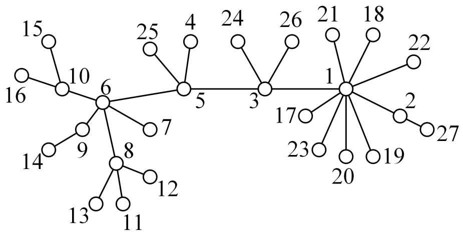
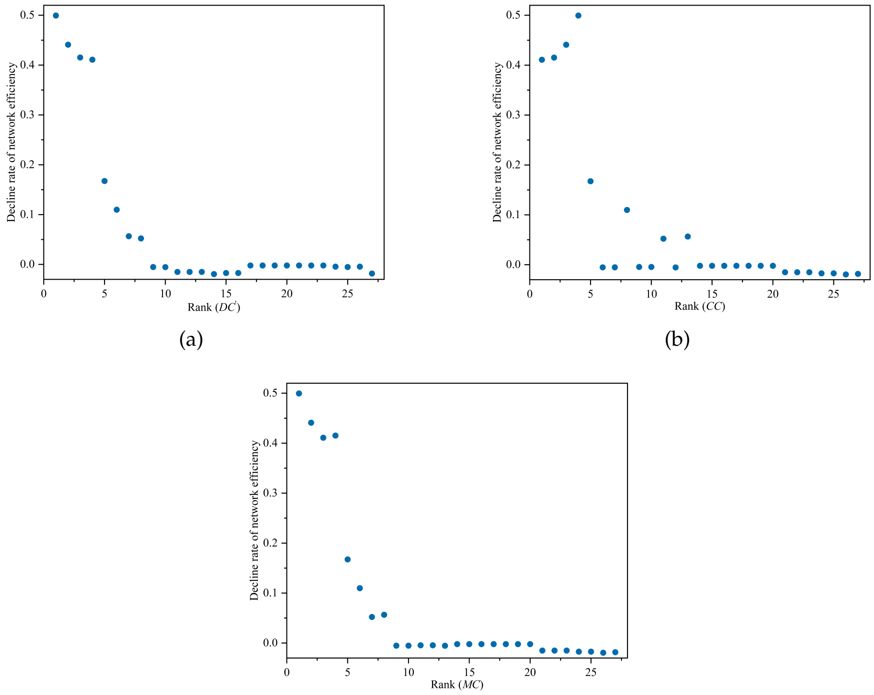


Figure 1. The network with 27 nodes and 26 edges.



(a)

(b)

(c)

Figure 2. The correlation between the decline rate of network efficiency and the importance of nodes ranked by $DC^{'}$, CC, and MC. (**a**) $DC^{'}$. (**b**) $CC^{'}$. (**c**) $MC^{'}$.

3.4. LPA Algorithm

First, we introduce the process of the LPA algorithm [12].

Step 1: We assign a label to each node in the network and the labels are different for different nodes, i.e., $l_i = i$, and if $i \neq j$, then $l_i \neq l_j$ $(i, j \in V(G))$.

Step 2: Randomly select node i from $V(G)$ and update the label of this node according to the following rule: Node i selects one of the most frequently occurring labels from the labels of its neighboring nodes as the new label of node i. If the label with the most frequent occurrence is not unique, one is chosen randomly.

Step 3: The algorithm stops when the labels of all nodes are no further updated.

In Step 2, when the node randomly selects labels it may be the case that a wrong label selection in one step leads to wrong label selection in each of the following steps. Poor quality communities may eventually be detected. In section 4.1.2, we will introduce a new rule to address this shortcoming.

4. The Proposed Algorithm (RWBS)

4.1. The Detailed Steps of Algorithm

The proposed algorithm (RWBS) consists of three main steps: random walk based on different seed nodes, propagation of labels and combination of communities. We explain these three steps in details below.

4.1.1. Random Walk Based on Different Seed Nodes

Let the transfer probability matrix is $P = (p_{ij})_{n\times n}$. In general, the probability of node i jumping to its neighboring nodes is the same, i.e., $p_{ij} = \frac{A_{ij}}{k(i)}$. For example, in Figure 3, node i has the same probability of selecting node u and node v as the node for the next jump. But the proposed algorithm changes the transfer rule of node i. We consider that nodes with larger degree have higher probability to attract node i, i.e., node i has higher possibility to transfer to node v in Figure 3 ($k(v) = 4 > k(u) = 1$). p_{ij} is defined as Eq. (10).

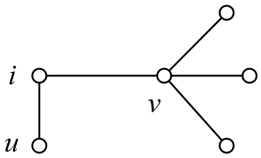


Figure 3. A network to explain the transfer probability of node i.

$$p_{ij} = \frac{k(j) * A_{ij}}{\sum\limits_{u\in N(i)} k(u)} \tag{10}$$

After the random walk algorithm is ended, the random walker prefers to stay at nodes with large degree. If the node with large degree is chosen as the initial node for random walk, there is no doubt that the random walker will stay at the node with large degree when the random walk stops. It will result that the information of the nodes with small degree will be ignored. Then, a new rule for the selection of initial nodes is proposed. To better obtain the structure of the network, we consider the distance between the nodes when selecting the initial nodes. The selected nodes with small mixed centrality tend to be located at the fringe of the network. Thus, when the selected seed nodes have the larger distance between them and have the small mixed centrality, the random walker can walk faster to the center nodes, and the structure of the network can be obtained faster. To select seed nodes with the above characteristics, we define the attraction between nodes $F(i, j)$, which considers the mixed centrality of nodes and $d(i, j)$.

$$F(i, j) = \frac{MC(i) + MC(j)}{2d(i, j)} \tag{11}$$

After obtaining the attraction $F(i, j)$ between any two nodes, we select the set of seed nodes S. For networks with ground-truth, we set the number of seed nodes is the number of communities, i.e. c. Let $community = \{community(i) : i \in V(G)\}$, which store the name of the community to which each node on the network belongs. The pseudo-code for selecting specific seed nodes is shown in Algorithm 1. For networks without ground-truth, we set the number of seed nodes is $n/4$, and the pseudo-code for selecting specific seed nodes is shown in Algorithm 2.

Algorithm 1 Select seed nodes on the network with ground-truth.

```
Input:
    $G = (V(G), E(G)); S = \{\}; community = \{community(i) : i \in V(G)\}$
Output:
    $S$
$(i, j) = \arg\min_{i,j}\{F(i, j) : i \neq j \in N(G)\};$    ▷ Randomly select one $(i, j)$ when $(i, j)$ is not unique.
if $community(i) \neq community(j)$ then
    $S \leftarrow S \cup \{i, j\};$
    $com\_node = find(community == community(i)) \cup find(community == community(j));$
    $V(G) \leftarrow V(G) - com\_node;$
else
    $S \leftarrow S \cup \{i\}$ (or $S \leftarrow S \cup \{j\}$);
    $com\_node = find(community == community(i))$ (or $com\_node = find(community == community(j))$)
    $V(G) \leftarrow V(G) - com\_node;$
end if
while $|S| < c$ do
    Initialize $M \leftarrow +\infty$ ;
    for $i = 1$ to $|V(G)|$ do
        Initialize $sum \leftarrow 0;$
        Let $Node \leftarrow S;$
        while $Node \neq \varnothing$ do
            Select $j \in Node;$
            $sum \leftarrow sum + F(i, j);$
            $Node \leftarrow Node - j$
        end while
        if $sum < M$ then
            $M \leftarrow sum;$
            $x \leftarrow i;$
        end if
    end for
    $S = S \cup \{x\};$
    $com\_node = find(community == community(x));$
    $V(G) \leftarrow V(G) - com\_node;$
end while
```

Let $\vec{v}(0)$ denote the probability that the random walker stays at each node in the initial state, and the i-th ($1 \leqslant i \leqslant n$) element of $\vec{v}(0)$ is $\vec{v}(0)_i$. The proposed algorithm lets the random walker start from the seed nodes (S) in the initial state, and if the degree of the node is larger, the probability of starting from that node is higher. It is shown in Eq. (12).

$$\vec{v}(0)_i = \begin{cases} \frac{k(i)}{\sum\limits_{j \in S} k(j)}, & i \in S \\ 0, & else \end{cases} \tag{12}$$

We use the random walk with restart algorithm (RWR) [46], which is a modification of the random walk algorithm. The algorithm faces two options at each step of the walk, the first option is to jump randomly to the neighbor node of the current node, and the other option is to return to the initial node, i.e., S. It contains a parameter α, which indicates the restart probability. When the number of random walk steps t is too large, the random walker prefers to stay at the node with large degree. The six degrees of separation theory states that the path length between any two people in the social network is short. That is, everyone in the social network can reach out to others in about six steps or less [47]. Thus, let the random walker stop after walking 6 steps, i.e., $t = 5$.

$$\vec{v}(t+1) = \alpha P^T \vec{v}(t) + (1-\alpha)\vec{v}(0),\ t = 0,1,2,...,5 \tag{13}$$

In this paper, let $\alpha = 0.96$. We can obtain the probability $\vec{v} = (v(1), v(2), ..., v(n))^T$ that the random walker stays at each node after the end of the random walk algorithm.

Algorithm 2 Select seed nodes on the network without ground-truth.

```
Input:
    $G = (V(G), E(G)); S = \{\};$
Output:
    $S$
$(i,j) = \arg\min_{i,j}\{F(i,j) : i \neq j \in V(G)\};$    ▷ Randomly select one $(i,j)$ when $(i,j)$ is not unique.
$S \leftarrow S \cup \{i,j\};$
$V(G) \leftarrow V(G) - S;$
while $|S| < \frac{n}{4}$ do
    Initialize $M \leftarrow +\infty;$
    for $i = 1$ to $|V(G)|$ do
        Initialize $sum \leftarrow 0;$
        Let $Node \leftarrow S;$
        while $Node \neq \varnothing$ do
            Select $j \in Node;$
            $sum \leftarrow sum + F(i,j);$
            $Node \leftarrow Node - j$
        end while
        if $sum < M$ then
            $M \leftarrow sum;$
            $x \leftarrow i;$
        end if
    end for
    $S = S \cup \{x\};$
    $V(G) \leftarrow V(G) - \{x\};$
end while
```

4.1.2. Propagation of Labels

In this model we also consider the triangular to measure the closeness between node pairs. Its specific structure is shown in Figure 4. If node i and node j have a common neighbor node k, the triangle is formed between i, j, k, which indicates that they have a close relationship. We measure the closeness between node i and node j by the number of neighbor nodes between them. The number of triangles can be expressed by Eq. (14).

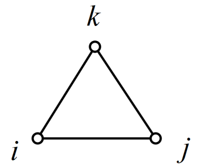


Figure 4. The structure of the triangle.

$$T(i,j) = |N(i) \cap N(j)|, (i,j) \in E(G) \tag{14}$$

doi:10.20944/preprints202407.0562.v1

where $T(i,j)$ denotes the number of triangular structures formed by node i and node j. The closeness $\rho(i,j)$ between node i and node j can be expressed by Eq. (15).

$$\rho(i,j) = \frac{1}{e^{-T(i,j)}+1}, (i,j) \in E(G) \tag{15}$$

For the label propagation in the next, we add a weight $w(i,j)$ to edge (i,j) ($(i,j) \in E(G)$), which considers the closeness between node i and node j and the probability of staying at node i and node j. $w(i,j)$ can be defined by Eq. (16).

$$\begin{aligned} w(i,j) &= \rho(i,j)\frac{v(i)+v(j)}{2} \\ &= \frac{v(i)+v(j)}{2(e^{-T(i,j)}+1)} \end{aligned} \tag{16}$$

Before performing label propagation, we assign a label to each node. The set of labels of nodes is set to $l = \{l_1, l_2, ..., l_n\}$, and $l_i \neq l_j$ when $i \neq j$. The label of the current node is updated according to the labels of its neighboring nodes in turn, and the label update rule is shown below.

$$l_i^{new} = \arg\max_{l_j}\{w(i,j) : j \in N(i)\} \tag{17}$$

where l_i^{new} denotes the new label of node i. When there exists more than one node j such that $w(i,j)$ has a maximum value, then l_i^{new} randomly selects one from the labels of these nodes. The algorithm stops when the labels of all nodes are no further updated. Let the final community as $Com = \{C_1, C_2, ..., C_c\}$. Where each node in C_i $(i = 1, 2, ..., c)$ has the same label. The label propagation rule in Eq. (17) solve the shortcoming of LPA mentioned in Section 3.4.

4.1.3. Combination of Communities

To improve the quality of the detected communities, we use Q as the optimization function to combine communities. Take any two different communities from Com and the rule of combining communities is shown in Eq. (18).

$$Com = \begin{cases} Com - C_i - C_j + (C_i \cup C_j), & \triangle Q(C_i \cup C_j) > 0 \\ Com, & else \end{cases} \tag{18}$$

where $\triangle Q = Q_2 - Q_1$, and Q_1 (Q_2) represents the modularity before (after) combining communities.

4.2. Time Complexity

The pseudo-code of RWBS is shown in Algorithm 3.

Obtain the mixed centrality of nodes on the network, and its time complexity is $O(n)$. Next, calculate the similarity $F(i,j)$ between any two nodes on the network, which has the time complexity $O(n^2)$. The seed nodes of random walk are obtained by Algorithm 1 or Algorithm 2. For networks with ground-truth, the time complexity of Algorithm 1 is $O(cn)$. For networks without ground-truth, the time complexity of Algorithm 2 is $O(2*(n-2)+3*(n-3)+...+\frac{n}{4}*\frac{3n}{4}) \approx O(n^2)$. The time complexity of obtaining the probability vector $\vec{v}$ after t steps is $O(tm)$ [10]. In this paper, the random walker is set to walk 6 steps on the network, so the time complexity of the process is $O(m)$. Assume that the label propagation requires h iterations to converge. The time complexity of the label propagation is $O(h*max(n,m))$. After the end of label propagation, assume that c communities are obtained. The time complexity of combining any two communities is $O(c^2)$. In summary, for networks with ground-truth, the total time complexity of the RWBS algorithm is $O(n^2)+O(cn)+O(n)+O(m)+O(h*max(n,m))+O(c^2) \approx O(n^2)$. For networks without ground-truth, the total time complexity of the RWBS algorithm is $O(n^2)$. So the time complexity of the RWBS algorithm is $O(n^2)$.

doi:10.20944/preprints202407.0562.v1

Algorithm 3 RWBS

```
Input:
    $G = (V(G), E(G))$;
    The undirected network $G$ with $n$ nodes and $m$ edges;
Output:
    $Com$;
Obtain the mixed centrality of all the nodes;
Get $F$ by Eq. (11);
Select the set of seed nodes $S$ according to Algorithm 1 or Algorithm 2;
Initialize $\vec{v}(0)$;
while $t < 6$ do
    Run the restart random walk algorithm according to Eq. (13);
end while
while $E(G) \neq \varnothing$ do
    Choose $(i, j) \in E(G)$;
    Calculate the weight $w(i, j)$ of edge $(i, j)$ by Eq. (16);
    $E(G) \leftarrow E(G) - (i, j)$;
end while
Assign a mutually different label to each node and let the set of labels is $L(0)$.
while $L(\lambda) \neq L(\lambda - 1)$ do
    while $V(G) \neq \varnothing$ do
        Select node $i \in V(G)$;
        Update the label of node $i$ according to Eq. (17);
        $V(G) \leftarrow V(G) - i$;
    end while
end while
Obtain $Com$ according to the label;
for $i = 1$ to $|Com| - 1$ do
    for $j = i + 1$ to $|Com|$ do
        if $\triangle Q(C_i \cup C_j) > 0$ then
            $Com \leftarrow Com - C_i - C_j + (C_i \cup C_j)$;
        end if
    end for
end for
Obtain the final community structure $Com$.
```

4.3. A Simple Example

To explain the RWBS algorithm in details, we give an sample network to illustrate how it detects communities. The sample network is shown in Figure 5(a), which contains 9 nodes and 15 edges. Figure 5(b) shows the real community structure of this sample network, which contains two communities $C_1 = [1, 2, 3, 4, 5]$ and $C_2 = [6, 7, 8, 9]$. The degree, closeness, and mixed centrality of the nodes are calculated, and the results are shown in Table 2.

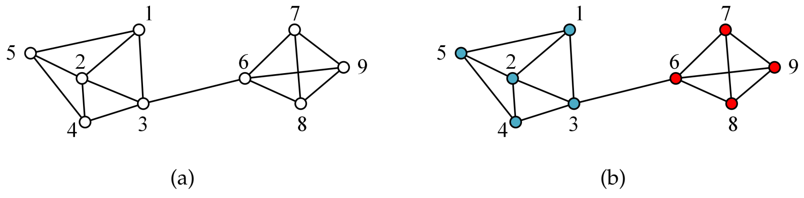


Figure 5. (**a**) The sample network with 9 nodes and 15 edges. (**b**) The real community structure of the sample network.

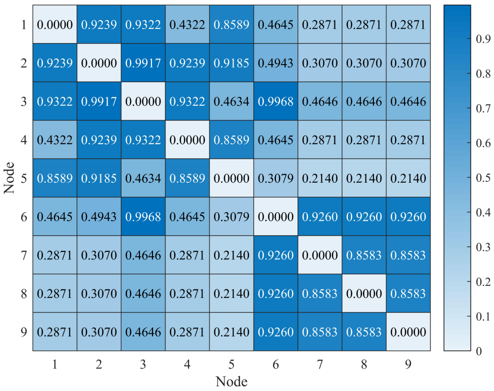
Table 2. The information of nodes

Information	1	2	3	4	5	6	7	8	9
$k(i)$	3	4	4	3	3	4	3	3	3
$DC^{'}(i)$	0.3750	0.5000	0.5000	0.3750	0.3750	0.5000	0.3750	0.3750	0.3750
$CC(i)$	0.0625	0.0667	0.0833	0.0625	0.0500	0.0769	0.0556	0.0556	0.0556
$MC(i)$	0.8643	0.9835	1.0000	0.8643	0.8536	0.9936	0.8583	0.8583	0.8583

First, the transfer probability matrix P can be obtained by Eq. (10), and it is shown in Eq. (19).

$$P = \begin{pmatrix} 0 & \frac{4}{11} & \frac{4}{11} & 0 & \frac{3}{11} & 0 & 0 & 0 & 0 \\ \frac{3}{13} & 0 & \frac{4}{13} & \frac{3}{13} & \frac{3}{13} & 0 & 0 & 0 & 0 \\ \frac{3}{14} & \frac{2}{7} & 0 & \frac{3}{14} & 0 & \frac{2}{7} & 0 & 0 & 0 \\ 0 & \frac{4}{11} & \frac{4}{11} & 0 & \frac{3}{11} & 0 & 0 & 0 & 0 \\ \frac{3}{10} & \frac{2}{5} & 0 & \frac{3}{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{4}{13} & 0 & 0 & 0 & \frac{3}{13} & \frac{3}{13} & \frac{3}{13} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{5} & 0 & \frac{3}{10} & \frac{3}{10} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{5} & \frac{3}{10} & 0 & \frac{3}{10} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{5} & \frac{3}{10} & \frac{3}{10} & 0 \end{pmatrix} \tag{19}$$

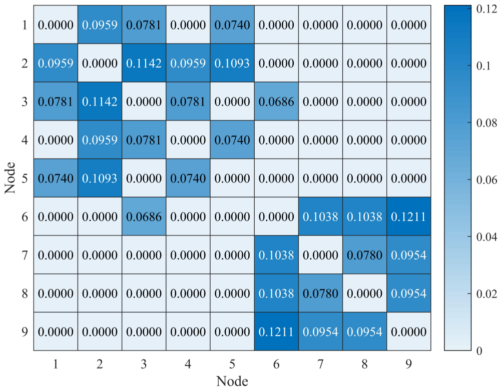
The similarity between nodes can be calculated by Eq. (11), and it is shown in Figure 6. Because the sample network contains two communities, let the number of seed nodes is 2, and obtain the specific seed nodes by Algorithm 1. From Figure 6, the similarity between node 5 and nodes 7, 8, 9 all have minimum value, node 5 can be selected as seed node. Moreover, we randomly select node 9 between 7, 8, 9 as seed node. Finally, $S = \{5, 9\}$.



Node	1	2	3	4	5	6	7	8	9
1	0.0000	0.9239	0.9322	0.4322	0.8589	0.4645	0.2871	0.2871	0.2871
2	0.9239	0.0000	0.9917	0.9239	0.9185	0.4943	0.3070	0.3070	0.3070
3	0.9322	0.9917	0.0000	0.9322	0.4634	0.9968	0.4646	0.4646	0.4646
4	0.4322	0.9239	0.9322	0.0000	0.8589	0.4645	0.2871	0.2871	0.2871
5	0.8589	0.9185	0.4634	0.8589	0.0000	0.3079	0.2140	0.2140	0.2140
6	0.4645	0.4943	0.9968	0.4645	0.3079	0.0000	0.9260	0.9260	0.9260
7	0.2871	0.3070	0.4646	0.2871	0.2140	0.9260	0.0000	0.8583	0.8583
8	0.2871	0.3070	0.4646	0.2871	0.2140	0.9260	0.8583	0.0000	0.8583
9	0.2871	0.3070	0.4646	0.2871	0.2140	0.9260	0.8583	0.8583	0.0000

Figure 6. The value of $F(i, j)$.

We assign a data pair (v_i, l_i) to each node. v_i denotes the name of the node. Let $\vec{v}(0) = (0, 0, 0, 0, \frac{1}{2}, 0, 0, 0, \frac{1}{2})^T$. $\vec{v}$ and w can be obtained when the algorithm ends (Four decimal places of $\vec{v}$ are retained). Where $\vec{v} = (0.0861, 0.1318, 0.1275, 0.0861, 0.1164, 0.1471, 0.0886, 0.0886, 0.1279)^T$, and w is shown in Figure 7. The label propagation process is shown in Figure 8. From Figure 8, the sample network contains two communities ($C_1 = [1, 2, 3, 4, 5]$ and $C_2 = [6, 7, 8, 9]$), which is consistent with the real community structure of this network. The modularity Q of this situation is 0.4244. If C_1 and C_2 are combined into one community, then $\Delta Q = -0.4244 < 0$. Thus, C_1 and C_2 are the final community structure.



Node	1	2	3	4	5	6	7	8	9
1	0.0000	0.0959	0.0781	0.0000	0.0740	0.0000	0.0000	0.0000	0.0000
2	0.0959	0.0000	0.1142	0.0959	0.1093	0.0000	0.0000	0.0000	0.0000
3	0.0781	0.1142	0.0000	0.0781	0.0000	0.0686	0.0000	0.0000	0.0000
4	0.0000	0.0959	0.0781	0.0000	0.0740	0.0000	0.0000	0.0000	0.0000
5	0.0740	0.1093	0.0000	0.0740	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0686	0.0000	0.0000	0.0000	0.1038	0.1038	0.1211
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.1038	0.0000	0.0780	0.0954
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.1038	0.0780	0.0000	0.0954
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.1211	0.0954	0.0954	0.0000

Figure 7. The value of w when $S = \{5, 9\}$.

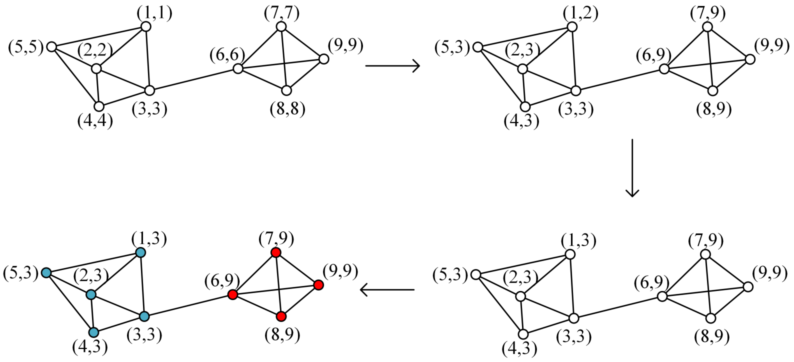


Figure 8. The label propagation process when $S = \{5, 9\}$.

5. Experiments

To evaluate the performance of RWBS algorithm in finding communities, we conducted experiments on the real-world and synthetic networks, respectively. For both the real-world networks with-ground truth and the synthetic networks, the community structure of these networks is already known, NMI and ARI are used to measure the similarity between the communities detected by RWBS and the real community structure. Higher values of these metrics indicate better quality of communities. For real-world networks without ground truth, the modularity (Q) is used to measure the quality of the detected communities, since we are not sure about the community structure of such networks.

5.1. Experiment on Real-World Networks

The following seven different real-world networks are chosen. The Karate, Dolphin, Political, and Football networks are networks with ground-truth. The Last, PGP, and Email networks are networks without ground-truth. In Table 3, we give information about these networks.

Table 3. The information of real networks

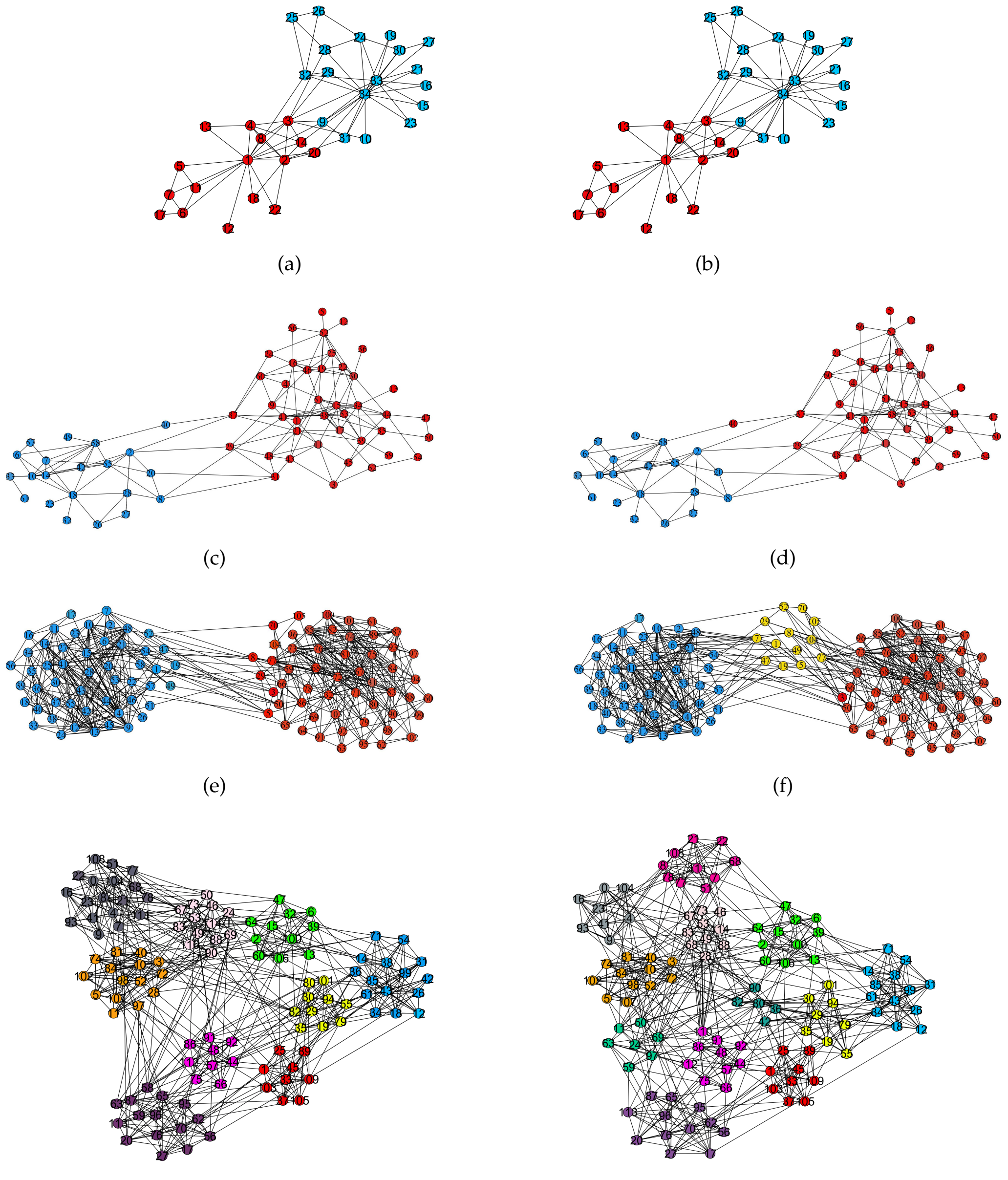
Network	n	m	k_{max}	$< k >$	c
Karate	34	78	17	4.588	2
Dolphin	62	159	12	5.129	2
Political	105	441	25	8.400	3
Football	115	616	12	10.661	12
Last	8003	16824	46	4.204	-
PGP	$10K$	$24K$	206	4.558	-
Email	$33K$	$180K$	1383	10.732	-

1. Karate network [48]: It is a social network with 34 members and 78 member relationships constructed by Zachary by observing the social relationships between members of a karate club at a university in the USA. Two members are considered to have edges to each other if they are frequently seen together in settings other than club activities. The club split into 2 smaller clubs with their own core because of a dispute between the director and the coach.
2. Dolphin network [49]: In 2003, Lusseau et al. observed the habits of 62 broad-snouted dolphins, and they found that these dolphins showed specific patterns of interactions and constructed a social network containing 62 nodes. If two dolphins are frequently active together, an edge exists between the two corresponding nodes in the network.
3. Political network [50]: The network is built by Krebs from pages of American politics-related books sold on Amazon. Its nodes represent American politics-related books, and edges represent a certain number of readers who have purchased both books. The nodes on the network are categorized as "liberal", "conservative" and "centrist". These divisions were manually analyzed by Newman based on the views and ratings of books on Amazon.
4. Football network [5]: The network shows games played in the American College Football League. The nodes in the network represent 115 teams, and edges represent two teams that have played a game against each other.
5. Last network [51]: This is a network based on the friendship between Last.fm users in Finland.
6. PGP network [52]: The network is an undirected network of bidirectional trust connections where each node contains both public and private keys.
7. Email network [53]: The network is built on the basis of relationships between users who email each other.

NMI and ARI are used to measure the quality of the communities, which are detected by RWBS on the network with ground-truth. The number of seed nodes affects the detected community structure for different networks. The results of the comparison between the communities detected by RWBS and the real communities of the network are shown in Figure 9.

For the Karate network, it contains two communities, so let the number of seed nodes be 2. In this case, $NMI = 1$ and $ARI = 1$. As can be seen from Figure 9, the community detected by RWBS at this case is consistent with its real community structure. For the Dolphin network, when the number of seed nodes is 2, $NMI = 0.8889$ and $ARI = 0.9348$. RWBS only divides node 40 to the wrong community and divides the other nodes to the correct community. This is almost consistent with the real community structure of the Dolphin network. For the Political network, it contains three communities, so let the number of seed nodes be 3. In this case, $NMI = 0.7365$ and $ARI = 0.7648$. The real community structure of the network is shown in Figure 9(f), which contains three communities. Although the proposed algorithm detects the number of communities of this network is two, the quality of the obtained communities is satisfactory. For the Football network, the number of seed nodes is 12, $NMI = 0.8671$ and $ARI = 0.7530$. RWBS detected that it contains 9 communities, and the division result is consistent with its real community structure. In summary, RWBS can find satisfactory communities when the number of seed nodes is c for networks with ground-truth.

To further show the superiority of RWBS in detecting communities, we selected Walktrap [10], LPA [12], KL [13], SLPA [31], CDME [34], and RIIM [17] as benchmark algorithms. Table 4 and Figure 10 show the results of the comparison between RWBS and benchmark algorithms in terms of NMI and ARI. From Table 4 and Figure 10, it can be seen that the results on the networks with ground-truth obtained by RWBS have maximum values in terms of NMI and ARI. Moreover, the division result obtained by RIIM on the Karate network is consistent with its real community structure. CDME (LPA and CDME) can also achieve satisfactory results on the Karate (Football) network. Walktrap detects poor community structures on these four networks.



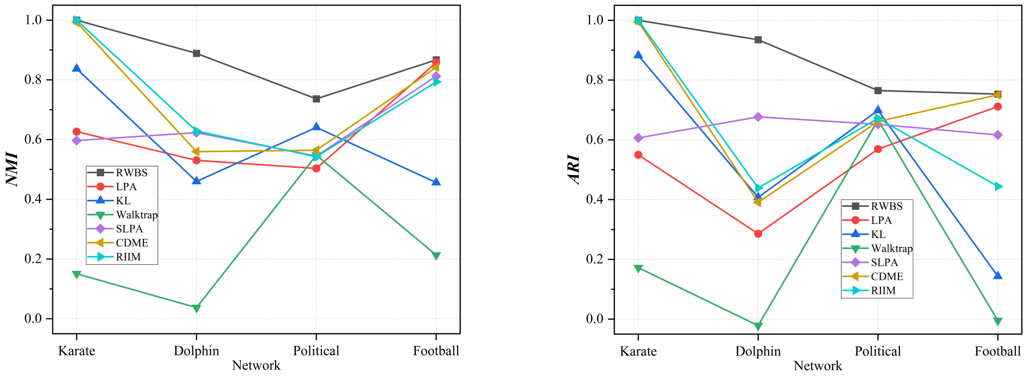
(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9. Comparison results obtained by RWBS with the real division result on real-world networks with ground-truth. (**a**) The division result obtained by RWBS ($NMI = 1$, $ARI = 1$, $c = 2$). (**b**) The real community structure of the Karate network ($c = 2$). (**c**) The division result obtained by RWBS (NMI=0.8889, $ARI = 0.9348$, $c = 2$). (**d**) The real community structure of the Dolphin network ($c = 2$). (**e**) The division result obtained by RWBS (NMI=0.7365, $ARI = 0.7648$, $c = 2$). (**f**) The real community structure of the Political network ($c = 3$). (**g**) The division result obtained by RWBS (NMI=0.8671, $ARI = 0.7530$, $c = 9$). (**h**) The real community structure of the Football network ($c = 12$).

We use Q as the metric to measure the quality of detected communities on networks without ground-truth. We let the number of seed nodes be $n/4$. Comparison results between RWBS and benchmark algorithms in terms of Q are shown in Table 5 and Figure 11. RWBS can obtain the maximum Q values on Last and Email networks, CDME can also achieve satisfactory Q values on both networks. For PGP network, RIIM can obtain the maximum Q value. RWBS, LAP, and CDME can also obtain large Q (they all have Q values more than 0.7).

Table 4. Comparison results on real networks with ground-truth (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently).

Approaches	Karate		Dolphin	
	NMI	*ARI*	*NMI*	*ARI*
RWBS	**1**	**1**	**0.8889**	**0.9348**
LPA	0.6268±0.2134	0.5499±0.2500	0.5305±0.0450	0.2857±0.0255
KL	0.8372	0.8823	0.4599	0.4077
Walktrap	0.1507	0.1722	0.0374	-0.0213
SLPA	0.5967±0.3174	0.6066±0.3646	0.6235±0.0873	0.6769±0.0776
CDME	0.9931±0.0218	0.9945±0.0175	0.5597±0.0493	0.3911±0.0576
RIIM	**1**	**1**	0.6287	0.4391
Approaches	**Political**		**Football**	
	NMI	*ARI*	*NMI*	*ARI*
RWBS	**0.7365**	**0.7648**	**0.8671**	**0.7530**
LPA	0.5033±0.0207	0.5690±0.0673	0.8582±0.0089	0.7113±0.0304
KL	0.6409	0.6987	0.4560	0.1437
Walktrap	0.5478	0.6661	0.2132	-0.0050
SLPA	0.5448±0.0142	0.6518±0.0238	0.8124±0.0286	0.6165±0.0679
CDME	0.5644±0.0208	0.6615±0.0279	0.8422±0.0364	0.7506±0.1306
RIIM	0.5427	0.6719	0.7931	0.4442



(a) (b)

Figure 10. Comparison results between RWBS and benchmark algorithms on networks with ground-truth in terms of *NMI* and *ARI* (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently). (**a**) *NMI*. (**b**) *ARI*.

Table 5. Comparison results on real networks without ground-truth (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently).

Approaches	Last		PGP		Email	
	c	*Q*	*c*	*Q*	*c*	*Q*
RWBS	183	**0.7671**	197	0.7094	62	**0.4102**
LPA	1574±8	0.6574±0.0017	1967±16	0.7425±0.0056	1686±27	0.3250±0.0141
KL	2	0.4017	2	0.4290	2	0.2703
Walktrap	1358	0.4783	1753	0.4335	353	0.3273
SLPA	1251±17	0.5988±0.0042	1629±16	0.6947±0.0038	525±23	0.3926±0.0123
CDME	627±42	0.7599±0.0084	580±44	0.7353±0.0074	312±2	0.4001±0.0116
RIIM	606	0.4457	560	**0.7625**	521	0.3014

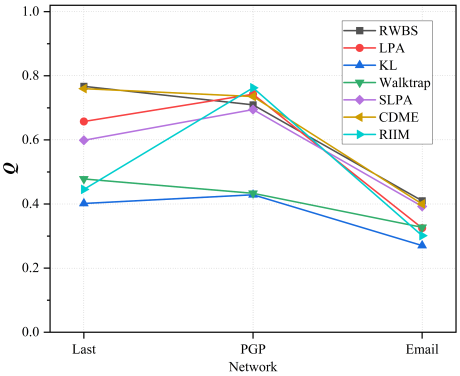


Figure 11. Comparison results on networks without ground-truth in terms of Q (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently).

5.2. Experiment on Synthetic Networks

We synthesize networks with the community structure by the LFR model [54]. Next, we further test the effectiveness of RWBS on these synthesized benchmark networks. Parameters of this model are shown in Table 6.

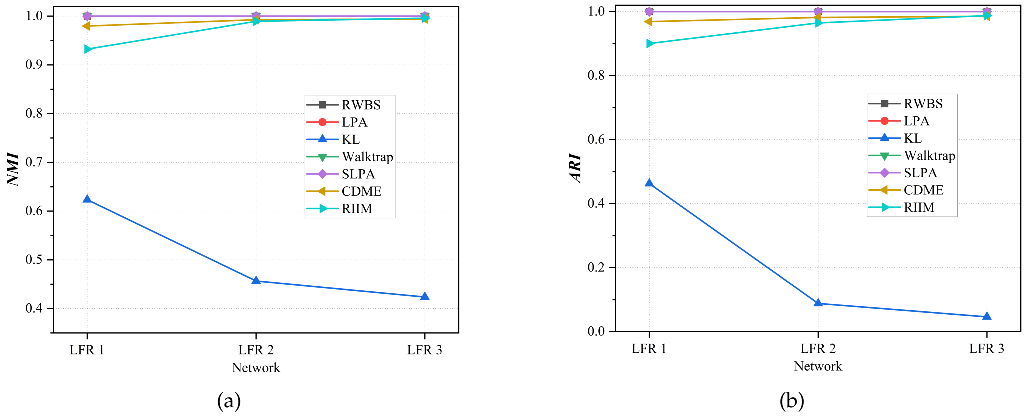
Table 6. Parameters of LFR

Networks	n	k_{max}	$< k >$	C_{min}	C_{max}	μ
LFR 1	100	20	10	20	30	0.1
LFR 2	1000	50	30	40	50	0.1
LFR 3	2000	50	30	45	50	0.1

We use RWBS and the benchmark algorithm to detect the community, and use NMI and ARI to measure the quality of the detected communities. The comparison results are shown in Table 7 and Figure 12. The community structures detected by RWBS, LPA, Walktrap, and SLPA on these three networks are completely consistent with their real community structures. The community structures detected by CDME and RIIM do not completely consistent with their real community structures, but satisfactory results can still be obtained. KL detected poor communities on these three networks.

Table 7. Comparison results on synthetic networks (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently).

Approaches	**LFR 1**		**LFR 2**		**LFR 3**	
	NMI	ARI	NMI	ARI	NMI	ARI
RWBS	**1**	**1**	**1**	**1**	**1**	**1**
LPA	**1**	**1**	**1**	**1**	**1**	**1**
KL	0.6233	0.4627	0.4565	0.0881	0.4237	0.0462
Walktrap	**1**	**1**	**1**	**1**	**1**	**1**
SLPA	**1**	**1**	**1**	**1**	**1**	**1**
CDME	0.9798±0.0596	0.9689±0.0948	0.9927±0.0206	0.9819±0.0537	0.9943±0.0169	0.9856±0.0430
RIIM	0.9325	0.9004	0.9894	0.9648	0.9959	0.9877



(a)

(b)

Figure 12. Comparison results between RWBS and benchmark algorithms on synthetic networks in terms of NMI and ARI (The values of LPA, SLPA, and CDME are averages obtained by running 10 experiments independently). (**a**) *NMI*. (**b**) *ARI*.

6. Results and Discussion

In this study, we propose a new random walk algorithm based on different seed nodes for community detection, named RWBS. The mixed centrality MC is proposed to measure the importance of nodes. According to the value of MC, the similarity $F(i,j)$ between nodes is calculated. Let the number of seed nodes of networks with (without) ground-truth is c $(n/4)$. For networks with and without ground-truth, based on $F(i,j)$, we propose algorithms to obtain seed nodes of the random walk, respectively. Furthermore, RWBS changes the transition probability matrix of the random walk algorithm and sets the number of steps of random walk to 6 according to the six degrees separation theory. We use a new label propagation rule that lets labels be updated in a fixed direction. Finally, modularity (Q) is chosen as an optimization function to combine communities, which can optimize the structure of communities. Experimental results on the network also verify the superiority of the RWBS algorithm.

We hope to generalize the RWBS algorithm to other networks in the future, such as directed networks, signature networks, weighted networks, etc. Moreover, we also hope to discover other strategies to find random walk seed nodes for better community division results.

Author Contributions: Writing—original draft, W.L.; validation, J.C. and X.Z.; writing—review and editing, J.C.; methodology, J.C. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NSFC grant numbers 12071351, 11971311 and 12161141003 and STCSM grant number 22JC1403602.

Data Availability Statement: The data of the Karate, Dolphin, Political, and Football networks can be downloaded from http://www-personal.umich.edu/~mejn/netdata/. The data of the Last, PGP, and Email networks can be downloaded from https://icon.colorado.edu/#!/networks and https://networkrepository.com/index.php.

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

References

1. Watts, D.-J.; Strogatz, S.-H. Collective dynamics of small world networks. *Nature* **1998**, *393(6684)*, 440–442.
2. Barabási, A.-L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286(5439)*, 509–512.
3. Barabási, A.-L. Scale-free networks: a decade and beyond. *Science* **2009**, *325(5939)*, 412–413.
4. Newman, M.E.J. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci.* **2001**, *98*, 404–409.
5. Girvan, M.; Newman, M.E.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **2002**, *99*, 7821–7826.
6. Scott, J. Social Network Analysis, 4th Edition.; SAGE Publications: London, UK, **2017**.
7. Arularasan, A.N.; Suresh, A.; Seerangan, K. Identification and classification of best spreader in the domain of interest over the social networks. *Cluster Comput.* **2018**, *3*, 1–11.

8. Redner, S. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B* **1998**, *4*, 131–134.
9. Guimerà, R.; Nunes Amaral, L. Functional cartography of complex metabolic networks. *Nature* **2005**, *433*, 895–900.
10. Pons, P.; Latapy, M. Computing Communities in Large Networks Using Random Walks. *Computer and Information Sciences* **2005**, *3733*, 284–293.
11. Jiao, Q.J.; Huang, Y.; Shen, H.B. Community mining with new node similarity by incorporating both global and local topological knowledge in a constrained random walk. *Physica A: Statistical Mechanics and its Applications* **2015**, *424*, 363–371.
12. Raghavan, U.N.; Albert, R.; Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **2007**, *76*, 036106.
13. Kernighan, B.W.; Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* **1970**, *49*, 291–307.
14. Barnes, E.R. An Algorithm for Partitioning the Nodes of a Graph. *SIAM Journal on Algebraic Discrete Methods* **1982**, *3*, 541–550.
15. Fortunato, S. Community Detection in Graphs. *Physics Reports* **2010**, *486*, 75–174.
16. Dhumal, A.; Kamde, P.M. Survey on Community Detection in Online Social Networks. *International Journal of Computer Applications* **2015**, *121*, 35–41.
17. Ullah, A.; Wang, B.; Sheng, J.; Long, J.; Khan, N.; Ejaz, M. A novel relevance-based information interaction model for community detection in complex networks. *Expert Systems With Applications* **2022**, *196*, 116607.
18. Liu, X.; Murata, T. An Efficient Algorithm for Optimizing Bipartite Modularity in Bipartite Networks. *Journal of Advanced Computational Intelligence and Intelligent Informatics* **2010**, *14*, 408–415.
19. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **2006**, *103*, 8577–8522.
20. Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133.
21. Blondel, V.D.; Guillaume, J.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks, *J. Stat. Mech.* **2008**, P10008.
22. Guimerà, R.; Amaral, L.A.N. Functional cartography of complex metabolic networks. *Nature* **2005**, *433*, 895–900.
23. Boettcher, S.; Percus, A.G. Optimization with Extremal Dynamics. *Phys. Rev. Lett* **2001**, *86*, 5211.
24. Duch, J.; Arenas, A. Community detection in complex networks using extremal optimization. *Phys. Rev. E* **2005**, *72*, 027104.
25. M'Barek, M.B.; Borgi, A.; Bedhiafi, W.; Hmida, S.B. Genetic Algorithm for Community Detection in Biological Networks. *Procedia Computer Science* **2018**, *126*, 195–204.
26. Chen, K.; Bi, W. A new genetic algorithm for community detection using matrix representation method. *Physica A: Statistical Mechanics and its Applications* **2019**, *535*, 122259.
27. Pourabbasi, E.; Majidnezhad, V.; Afshord, S.T.; Jafari, Y. A new single-chromosome evolutionary algorithm for community detection in complex networks by combining content and structural information. *Expert Systems with Applications* **2021**, *186*, 115854.
28. Su, Y.; Zhou, K.; Zhang, X.; Cheng, R.; Zheng, H. A parallel multi-objective evolutionary algorithm for community detection in large-scale complex networks. *Information Sciences* **2021**, *576*, 374–392.
29. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The Pagerank Citation Ranking: Bring Order To The Web. Technical Report, Stanford University, **1998**.
30. Rosvall, M.; Bergstrom, C.T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **2008**, *105*, 1118–1123.
31. Xie, J.; Szymanski, B.K.; Liu, X. SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process. In 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, BC, Canada, IEEE, **2011**, 344–349.
32. Laassem, B.; Idarrou, A.; Boujlaleb, L.; Iggane, M. Label propagation algorithm for community detection based on Coulomb's law. *Physica A: Statistical Mechanics and its Applications* **2022**, *593*, 126881.
33. Fang, W.; Wang, X.; Liu, L; Wu, Z.; Tang, S.; Zheng, Z. Community detection through vector-label propagation algorithms. *Chaos, Solitons & Fractals* **2022**, *158*, 112066.

34. Sun, Z.; Sun, Y.; Chang, X.; Wang, Q.; Yan, X.; Pan, Z.; Li, Z. Community detection based on the Matthew effect. *Knowledge-Based Systems* **2020**, *205*, 106256.
35. Chen, X.; Hu, J.; Chen, Y. GBTM: Community detection and network reconstruction for noisy and time-evolving data. *Information Sciences* **2024**, *679*, 121069.
36. Danon, L.; Duch, J.; Diaz-Guilera, A.; Arenas, A. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* **2005**, P09008.
37. Rand, W.M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **1971**, *66*, 846–850.
38. Kim, Y.; Son, S.-W.; Jeong, H. Finding communities in directed networks. *Phys. Rev. E* **2010**, *81*, 016103.
39. Bonacich, P.F. Factoring and weighting approaches to status scores and clique identification, *J. Math. Sociol.* **1972**, *2*, 113–120.
40. Newman, M.E.J. A measure of betweenness centrality based on random walks. *Social Networks* **2005**, *27*, 39–54.
41. Freeman, L.C. Centrality in social networks conceptual clarification. *Social Networks* **1978**, *1*, 215–239.
42. Stephenson, K.; Zelen, M. Rethinking centrality: Methods and examples. *Social Networks* **1989**, *11*, 1–37.
43. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117.
44. Wasserman, S.; Faust, K. Social Network Analysis: Methods and Applications, first ed.: Cambridge University, New York, **1994**.
45. Ren, Z.; Shao, F.; Liu, J.; Guo, Q.; Wang, B. Node importance measurement based on the degree and clustering coefficient information. *Acta Phys. Sinica* **2013**, *62*, 128901.
46. Tong, H.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications, In Sixth International Conference on Data Mining, Hong Kong, China, IEEE, **2006**, 613–622.
47. Hua, J.; Yu, J.; Yang, M.S. Fast clustering for signed graphs based on random walk gap. *Social Networks* **2020**, *60*, 113–128.
48. Zachary, W.W. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **1997**, *33*, 452–473.
49. Lusseau, D.; Schneider, K.; Boisseau, O.; Haase, P.A.; Slooten, E.; Dawson, S. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **2003**, *54*, 396–405.
50. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113.
51. Toivonen, R.; Kovanen, L.; Kivelä, M.; Onnela, J.; Saramäki, J.; Kaski, K. A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks* **2009**, *31*, 240–254.
52. Boguñá, M.; Pastor-Satorras, R.; Díaz-Guilera, A.; Arenas, A. Models of social networks based on social distance attachment. *Phys. Rev. E* **2004**, *70*, 056122.
53. Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin Texas, AAAI Press, **2015**, 4292–4293.
54. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.