

Article

Community Detection Based on Node Influence and Similarity of Nodes

Yanjie Xu, Tao Ren *  and Shixiang Sun

Software College, Northeastern University, Shenyang 110169, China; 1710462@stu.neu.edu.cn (Y.X.); 1810476@stu.neu.edu.cn (S.S.)

* Correspondence: chinarentao@163.com or rent@swc.neu.edu.cn

Abstract: Community detection is a fundamental topic in network science, with a variety of applications. However, there are still fundamental questions about how to detect more realistic network community structures. To address this problem and considering the structure of a network, we propose an agglomerative community detection algorithm, which is based on node influence and the similarity of nodes. The proposed algorithm consists of three essential steps: identifying the central node based on node influence, selecting a candidate neighbor to expand the community based on the similarity of nodes, and merging the small community based on the similarity of communities. The performance and effectiveness of the proposed algorithm were tested on real and synthetic networks, and they were further evaluated through modularity and NMI analysis. The experimental results show that the proposed algorithm is effective in community detection and it is quite comparable to existing classic methods.

Keywords: complex network; community detection; node influence; similarity of nodes

MSC: 05C90



Citation: Xu, Y.; Ren, T.; Sun, S. Community Detection Based on Node Influence and Similarity of Nodes. *Mathematics* **2022**, *10*, 970. <https://doi.org/10.3390/math10060970>

Academic Editor: Alfredo Milani

Received: 20 January 2022

Accepted: 14 March 2022

Published: 18 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Complex networks play an important role in representing complex systems in subject areas such as social networks, biology, psychology, informatics, management, etc. [1]. Complex networks can be represented by a graph $G = (V, E)$, where V and E represent the set of nodes and edges in the network, respectively [2]. In real networks, nodes and edges can represent various individuals and relationships. For example, identifying influential nodes is one of the research hotspots in the study of complex networks, used to analyze the network structure [3,4]. However, using only a group of influential nodes does not enable one to grasp a network's hidden information completely. The community structure that exists in real networks can help us to analyze the network structure in depth [5]. It is generally believed that community structure is a network subgraph with close internal connections and sparse external connections, which may have a certain independent structure or specific function [6]. In reality, community detection is useful in solving numerous problems affecting human communities, such as analyzing networks of social opinion [7], recommending products for customers [8], finding users displaying malicious activity to protect system security [9], identifying influential nodes [10,11], and so on.

The question of how to detect a community is an important issue that attracts researchers from all over the world. There are signs of great progress in the research on community detection in complex networks, including GN [12], LPA [13], EM [14], and so on. GN is based on edge betweenness; LPA is based on label propagation; and EM is a clustering algorithm. However, these algorithms have certain limitations. The accuracy of the GN algorithm is high, but its complexity costs are much higher. LPA runs quickly, but its accuracy is unstable. EM displays a better ability to cluster, but it is unable to detect highly modular communities. Therefore, it is necessary to detect communities accurately and stably.

To address this problem, we propose a community detection algorithm named NINS (a community detection algorithm based on node influence and the similarity of nodes). In this paper, the NINS algorithm consists of three parts: identifying the central node, selecting a candidate neighbor to expand the community, and merging the community. The proposed algorithm is stable, and can detect modular and realistic communities. Experiments on real and LFR networks show that the proposed NINS is effective and quite comparable to existing community detection algorithms.

The structure of this paper is as follows: Section 2 reviews the related work. In Section 3, the proposed algorithm is introduced, including the detailed steps of the algorithm, complexity analysis, and a description of the implementation process. The network data descriptions and numerical results based on various methods applied to real and synthetic networks, respectively, are shown in Section 4. Moreover, the experimental results are discussed in Section 5. Finally, our conclusions are presented in Section 6.

2. Related Work

Community detection is a hot topic in network science. The earliest study on this subject was reported in 1970 [15]. Next, Girvan and Newman proposed the network community structure [12]. From that time until now, research on community detection has gradually developed. It has been proven that community detection is an NP-hard problem [16,17]. Some classic algorithms are listed in Table 1, where N is the number of nodes in the network, M is the number of edges in the network, and m is the number of iterations.

Table 1. Comparison of some classic community detection algorithms.

Name	Classification	Parameters	Complexity	Pros and Cons
GN [12]	divisive	-	$O(NM^2)$	High modularity but high complexity
LPA [13]	agglomerative	m	$O(N)$	Fast but needs the exact algorithm iteration number m
EM [14]	agglomerative	m	$O(mN^3)$	High complexity and low modularity
Louvain [18]	agglomerative	-	$O(N \log N)$	Higher modularity but unstable

From the perspective of detecting community structure, some algorithms (e.g., GN [12], Louvain [18], and CDIA [19]) detect non-overlapping communities, where each node only belongs to one community; other algorithms (e.g., CPM [20] and ONES [21]) can detect overlapping communities, where one node can belong to two or more communities. From the perspective of hierarchical clustering, these types of algorithms can be divided into two categories [16,22]. Some of them are agglomerative algorithms, where each node assigns as a community and iteratively merges the smaller communities according to their similarities [18,23,24]. Other hierarchical clustering algorithms are divisive methods, where the network is taken as a community and is divided into some smaller communities. GN [12] is the most popular divisive algorithm, assessing the central edges based on their shortest path centrality. However, it only accurately detects small or medium networks of 10,000 nodes at most. Thus, Arasteh et al. [25] proposed a fast divisive algorithm based on edge degrees. Chen et al. [26] detected communities in complex networks using an edge-deleting algorithm with restrictions.

From another perspective, the label propagation algorithm (LPA) [13] is also a famous community detection algorithm. In LPA, each node has a unique label, which is updated based on the most common labels among its neighbors. The complexity of LPA is $O(N)$. The convergence of LPA is provable mathematically but it requires the exact algorithm iteration number, which is always dependent on the network parameters (e.g., node and edge numbers) [27]. In 2019, Basuchowdhuri et al. [27] proposed a community detection

algorithm named LINCOS, which involves two steps: selecting the broker node by means of an objective function and merging this node into the community with the majority of its neighbors.

From the perspective of deep learning, Al-Andoli et al. [28] introduced a deep-learning algorithm for community detection. To decrease the trainable parameters needed for the deep-learning model, they first divided the network into some smaller parts and proposed a novel similarity constraint function that improved the algorithm's effectiveness. Agrawal and Patel [29] proposed a community detection algorithm named SAG based on topological structure and node attributes. Tsitsekis et al. [30] proposed a scalable community detection method for complex data graphs via hyperbolic network embedding and graph databases. He et al. [31] used the modularity function to sample node sequences and learn node representation by using the skip-gram model to detect communities.

In recent years, some other algorithms have been proposed to detect communities. For example, Feng et al. [32] proposed a community detection algorithm based on node betweenness and structure similarity. Majid Arasteh [33] proposed a gravity algorithm to detect the communities of large-scale networks; the proposed algorithm runs quickly but its accuracy is not ideal. Pourabbasi [34] proposed a new single-chromosome evolutionary algorithm for community detection in complex networks by combining content and structural information. Newman [35] proposed an information-theoretic method for discovering the building blocks in specific networks to show the consistency of community structure in complex networks. Cauteruccio et al. [36] proposed an algorithm to identify virtual communities based on user stereotypes. Mengoni et al. [37] proposed an algorithm to identify hidden communities based on history analysis and session analysis of co-occurrence of activities.

In a bid to further support and enhance the study, discussion, and understanding of community detection, systematic literature reviews have been performed to analyze community detection approaches. Naik [38] surveyed parallel and distributed paradigms for community detection in social networks. Yassine et al. [39] reviewed community detection methods using social network analysis in online learning environments. Attea et al. [40] performed a review of heuristics and metaheuristics for community detection. Huang et al. [41] summarized the community detection methods in multilayer networks. Calderer [42] reviewed community detection in large-scale bipartite biological networks. Gasparetti et al. [43] reviewed community detection in social recommender systems. Rosvall [44] provided a focused review of community detection methods with different motivations, including the cut-based perspective, clustering perspective, stochastic equivalence perspective, and dynamical perspective. Dao et al. [45] conducted a comparative evaluation of community detection methods.

3. Algorithm

In this paper, based on node influence and the similarity of nodes, an agglomerative-based community detection algorithm named NINS is proposed to detect modular communities by producing groups of densely connected nodes. The proposed algorithm works on unweighted and undirected networks and detects non-overlapping communities, the numbers of which do not need to be set before the execution of the algorithm. NINS consists of the following three essential steps: identifying the central node based on node influence, selecting a candidate neighbor to expand the community based on the similarity of nodes, and merging the small community based on the similarity of communities. Table 2 summarizes the symbols and notations used in the paper.

Table 2. Symbols and variables used herein.

Notation	Description
G	a graph represents network
V	set of nodes
E	set of edges
$\Gamma(i)$	neighbor set of node i
k_i	degree of node i
S_{ij}	the similarity of nodes i and j
$aveS(., i)$	the average similarity of node i with its neighbors
$\Gamma(i) \cap \Gamma(j)$	common neighbor set of node i and j
S	maximum number of nodes in the small community
C_i	the i -th community
n_q	the number of nodes that belong to the community C_j and are connected to the community C_i
N	number of nodes in the network
M	number of edges in the network
t	number of the small community
$\langle k \rangle$	average degree of node
m	number of iterations
$\langle d \rangle$	average distance of nodes
D	network diameter
C	clustering coefficient of the network
r	assortative coefficient of the network
Q	modularity
a_{ij}	represents whether nodes i and j are connected or not: if nodes i and j are not connected, $a_{ij} = 0$; otherwise, $a_{ij} = 1$
$\delta(C_i, C_j)$	represents whether nodes i and j are in the same community or not: If $\delta(C_i, C_j) = 1, C_i = C_j$; if $\delta(C_i, C_j) = 0, C_i \neq C_j$

3.1. Three Essential Steps of the Algorithm

3.1.1. Step 1: Identifying the Central Node Based on Node Influence

The central node has the highest influence in the network, which can attract its neighbors. In the network, the greater the degree of a node, the lesser it can be affected by one of its neighbors. For example, in the rumor-spreading process, the more neighbors a node has, the lesser it will be affected/influenced by one of its neighbors. Thus, $1/k_j$ is used to represent node i 's influence on its neighbor j . The influence of node i can be calculated as the sum of the influence on its neighbors:

$$I(i) = \sum_{j \in \Gamma(i)} \frac{1}{k_j} \tag{1}$$

where $\Gamma(i)$ is the neighbor set of node i and k_j is the degree of node j .

In step 1, we sort the nodes by node influence and choose the first node as the central node of the community.

3.1.2. Step 2: Expanding the Community Based on the Similarity of the Nodes

It is known that nodes within a community are more closely connected than those outside a community. In step 2, the similarity of the nodes and the average similarity of each node with its neighbors are used to select a candidate neighbor, which is used to expand the community. The AA [46] algorithm is selected to measure the similarity of the two nodes, which can better reflect the degree of node connection than the other node similarity algorithm based on local structure information:

$$S(i, j) = \sum_{t \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log k_t} \tag{2}$$

where t is the common neighbor of nodes i and j .

The average similarity is proposed as a measure to reflect the average similarity of each node with its neighbors, which can be calculated as follows:

$$aveS(., i) = \frac{1}{k_i} \sum_{j \in \Gamma(i)} S(i, j) \tag{3}$$

In step 2, node i 's neighbor j is added to the community when $aveS(., j) < S(i, j)$. The reason for this is that the similarity among nodes in the same community is larger than that of others that do not belong to that community. A higher similarity would ensure that the community consists of nodes with a more dense connection. The central node's neighbors may not always belong to the same community as it is the node with the highest similarity. The similarity of connected nodes in the same community is generally larger than the average similarity.

3.1.3. Step 3: Merging Small Communities Based on Community Similarity

In step 3, the community C_i with the maximum number of S nodes would be selected as a small community. It is known that the nodes within the same community are more closely connected. Therefore, it is obvious that the small community is likely to merge with the community with the highest similarity. Modularity maximization and label propagation are two methods used as community similarity measures, which are often used in optimization methods for detecting community structures in networks. However, it has been shown that modularity suffers from a resolution limit [47]; therefore, it is unable to detect small communities [46]. Thus, label propagation [13] is introduced to measure the similarity between small communities and the other communities:

$$S(C_i, C_j) = n_q(q \in C_j \cap \Gamma(t, t \in C_i)) \tag{4}$$

where n_q represents the number of nodes that belong to the community C_j and are connected to community C_i .

3.2. The Proposed Algorithm and Complexity

Firstly, the proposed algorithm calculates node influence and chooses the most influential node as the central node. Then, it selects a candidate neighbor to expand the community based on the similarity of nodes and average similarity (if the neighbor has one neighbor, it should be merged into the community). The process runs iteratively. Next, according to the number of nodes in the community, the initial community is divided into small and large communities. Finally, the small community is merged to the community with the highest number of neighbors. The algorithm is stopped when the number of nodes of each community in the network is greater than S . In this paper, $S = 3$. The pseudo-algorithm of NINS is shown in Algorithm 1. A real network, Karate [48], was selected to illustrate the algorithm.

The NINS algorithm takes $O(N\langle k \rangle)$ to calculate node influence and repeats this process N times in steps 1 and 2. According to the changes in the network, choosing the central node and selecting a candidate neighbor at each iteration takes $O(1 + \log N)$. In step 3, beginning with t small communities and merging these small communities into other communities ($t \leq N$) takes $O(t\langle k \rangle)$ time. The algorithm does not stop until the node number of communities is larger than S .

The complexity cost of the proposed algorithm is $O(N\langle k \rangle + N(1 + \log N) + St\langle k \rangle)$. When $n \log n > M$, the complexity of NINS is $O(N \log N)$; otherwise, it is $O(M)$. The complexity cost of GN [12] is $O(NM^2)$, that of Louvain [18] is $O(N \log N)$, LPA [13] is $O(N)$, CDIA [19] is $O(M)$, and EM [14] is $O(mN^3)$, where m is the number of iterations.

Algorithm 1: NINS Algorithm.

Input: Network $G = (V, E)$, maximum number of nodes in the small community S
Output: communities
for each i **in** $G.nodes$ **do**
 Calculate $I(i)$ by Equation (1)
end
Sort in descending order ($V, key = I(i)$)
for each i **in** $G.nodes$ that does not belong to a community **do**
 choose i with the highest influence as the central node
 for each j **in** $i.neighbors$ that does not belong to a community **do**
 Calculate the nodes similarity $S(i, j)$ by Equation (2)
 Calculate average nodes similarity $aveS(., j)$ by Equation (3)
 if node j has one neighbor or $S(i, j) > aveS(., j)$ **do**
 Merge node j into the community C_i that node i belongs to
 for each b **in** $j.neighbors$ that does not belong to a community **do**
 iterate 8–12
 end
 end
 end
 communities.append(C_i)
end
for each C_i **in** communities **do**
 if $n(C_i) \leq S$ **do**
 Calculate similarities of C_i with its neighbor communities by Equation (4)
 Merge C_i into the community with the highest similarity with C_i
 Remove C_i from communities and update communities
 end
end

3.3. Example of the Algorithm

A real network, Karate, was selected to illustrate the algorithm. The network of Karate is shown in Figure 1. The central node was first selected by sorting node influence based on Equation (1). Node 34 was the most influential node and its influence value was 5.767. Next, the similarity of the node and its neighbors was calculated. Then the community was expanded. The initial communities are shown in Table 3. Small and large communities were distinguished based on whether the node number of a community was greater than three or not. If the community node number was greater than three, it was considered a large community; otherwise, it was considered a small community. The initial results showed two small communities [10,17]. For node 10, only node 34 was its neighbor. It should thus be merged into the community C_1 , to which node 34 belongs. For node 17, its neighbors were node 6 and node 7. Both nodes 6 and 7 were in the community C_2 , and node 17 would be merged into community C_2 accordingly. The final community result of Karate offered by the proposed algorithm is shown in Figure 2, which was the same as the real community structure. It is worth noting that the size of nodes in the figure is positively correlated with node influence; that is, the greater influence of the node, the larger the node in the figure.

Table 3. The initial community structure of Karate.

Initial Result	Nodes of the Initial Community
C_1	34, 9, 31, 33, 24, 28, 34, 15, 16, 19, 21, 23, 27, 29, 30, 32, 25, 26
C_2	1, 2, 3, 4, 14, 5, 6, 7, 8, 11, 12, 13, 18, 20, 22
C_3	10
C_4	17

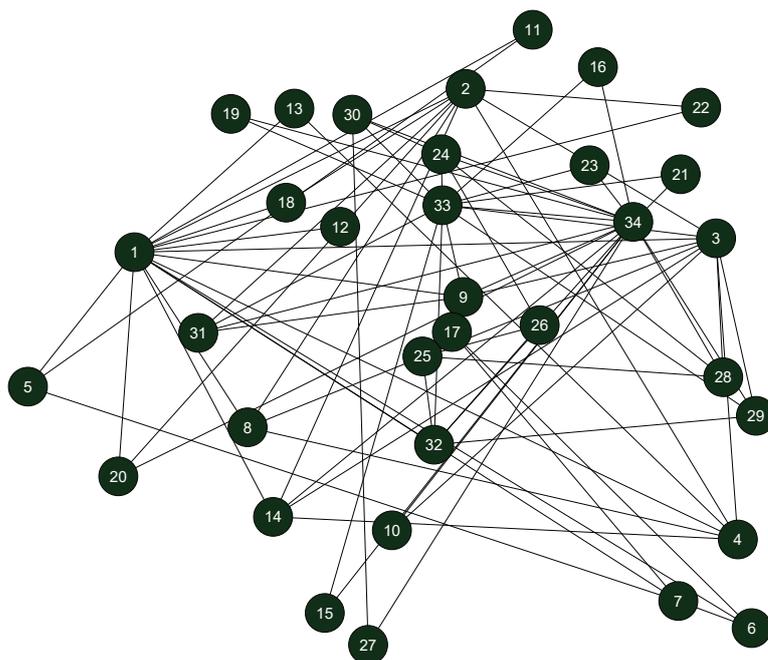


Figure 1. The Karate network.

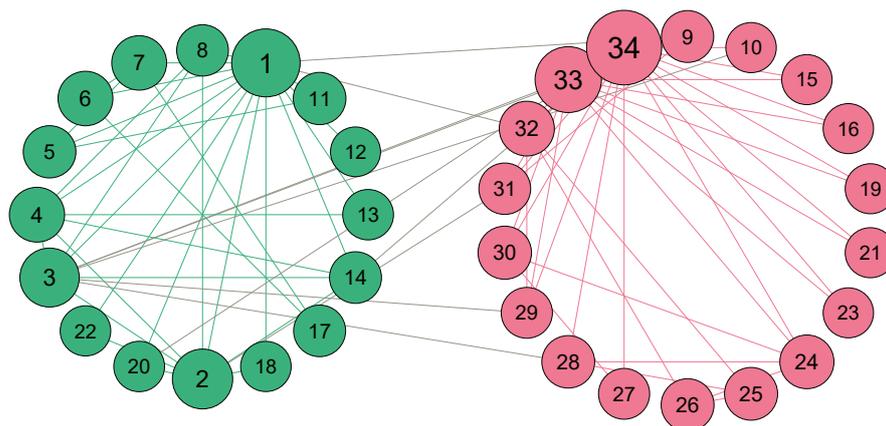


Figure 2. The final community structure of Karate according to the NINS algorithm.

4. Experiment

In this section, first, we describe the datasets used in our experiments. Then, we explain the evaluation criteria, referred to as modularity Q [23] and normalized mutual information (NMI) [49]. The results are explained at the end.

4.1. Data Description

To test the performance of NINS, seven real networks with different sizes were used for comparison with several classic methods. Dolphin [50] is a social network of 62 dolphins. Karate [48] is a real social network containing the network of friendships among the 34 members of a karate club at a US university. Football [12] is a network of American football games among Division IA colleges during the regular season of fall 2000. Course registration [19] is a record of college students at Northeastern University. NS [51] is a co-authorship network of scientists working on network science. Power [52] is the power grid of the western United States. Router [53] is a symmetrized snapshot of the structure of the Internet at the level of autonomous systems. Table 4 summarizes the key properties of the selected datasets.

Nine LFR networks [54] were generated and used to test the performance of NINS. The mixed parameter of the fixed network was 0.5, and the size of the community was $minc = 10$ and $maxc = 20$. The nodes of these LFR networks were 1000–9000, respectively.

Table 4. The basic topological features of five real networks.

Networks	N	M	$\langle k \rangle$	$\langle d \rangle$	D	C	r
Dolphin	62	159	2.5645	3.3570	8	0.2590	−0.0436
Karate	34	78	4.5882	2.4082	5	0.57.6	−0.4756
Football	115	613	10.6609	2.5082	4	0.4032	0.1624
Course registration	47	125	5.3191	3.2812	7	0.4883	0.1354
NS	379	914	4.8232	6.0419	17	0.7412	−0.0817
Power	4941	6594	2.6691	18.9892	46	0.0801	0.0035
Router	5022	6258	2.4922	6.4488	15	0.0116	−0.1384

4.2. Evaluation Criterion

The modularity Q [23] was used as one of the evaluation criteria to compare the performance of NINS with different algorithms on the considered real datasets.

$$Q = \frac{1}{2M} \sum_{i \neq j} (a_{ij} - \frac{k_i k_j}{2M}) \delta(C_i, C_j) \tag{5}$$

where $-1 \leq Q \leq 1$; M is the number of edges, a_{ij} represents whether nodes i and j are connected or not. If they are not connected, $a_{ij} = 0$; otherwise, $a_{ij} = 1$. C_i is the community that node i belongs to. $\delta(C_i, C_j)$ represents whether nodes i and j are in the same community or not. If $\delta(C_i, C_j) = 1$, this means that nodes i and j are in the same community, that is, $C_i = C_j$; if $\delta(C_i, C_j) = 0$, $C_i \neq C_j$.

The normalized mutual information (NMI) [47] was the other evaluation criterion used to determine the performance of the proposed NINS. It can be calculated as follows:

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} N_{ij} \log(\frac{N_{ij}N}{N_i N_j})}{\sum_{i=1}^{C_A} N_i \log(\frac{N_i}{N}) + \sum_{j=1}^{C_B} N_j \log(\frac{N_j}{N})} \tag{6}$$

where A is the real partition, B is the detected partition, C_A is the number of real communities, C_B is the number of detected communities, N_{ij} represents the number of nodes shared by real community i and the detected community j , the number of nodes is denoted as N , N_i is the sum over row i of matrix N_{ij} , and N_j is the sum over column j of matrix N_{ij} .

4.3. Experimental Performance of NINS

4.3.1. Experiment on Real Networks

The proposed algorithm was compared with several classic algorithms in seven real networks in terms of modularity Q and runtime. The results of the proposed NINS algorithm are shown in Table 5. For the Karate, Dolphin, Football and Course registration networks, the community results of the proposed algorithm are shown in Figures 2–5, respectively. The greater the influence of the node, the larger the node in the figure.

To verify the performance of NINS, comparison experiments were conducted with five different algorithms in five real networks. The modularity and runtime of six methods in five real networks are shown in Figures 6 and 7, respectively. As shown in Figure 6, the performance of the proposed NINS modularity was better than that of LPA [13] and EM [14] in five real networks, close to that of CDIA [19], and lower than that of GN [12] and Louvain [18]. As shown in Figure 7, the time complexity of the proposed NINS was better than that of the GN and EM algorithms, and close to that of the Louvain, LPA, and CDIA algorithms. The NMI of the communities found in the Karate network determined using our method was 1, that of GN was 0.0819, Louvain was 0.6782, LPA was 0.4213, CDIA

was 0.8372, and EM was 0.8372. The results obtained for the real networks indicate that the proposed NINS is effective in community detection and it is quite comparable to existing classic methods.

Table 5. The experimental results derived from performing NINS on real networks.

Networks	Q	Runtime(s)
Dolphin	0.4707	0.01
Karate	0.3715	0.007
Football	0.5684	0.095
Course registration	0.5005	0.012
NS	0.7774	0.092
Power	0.7586	3.4076
Router	0.8073	1.5912

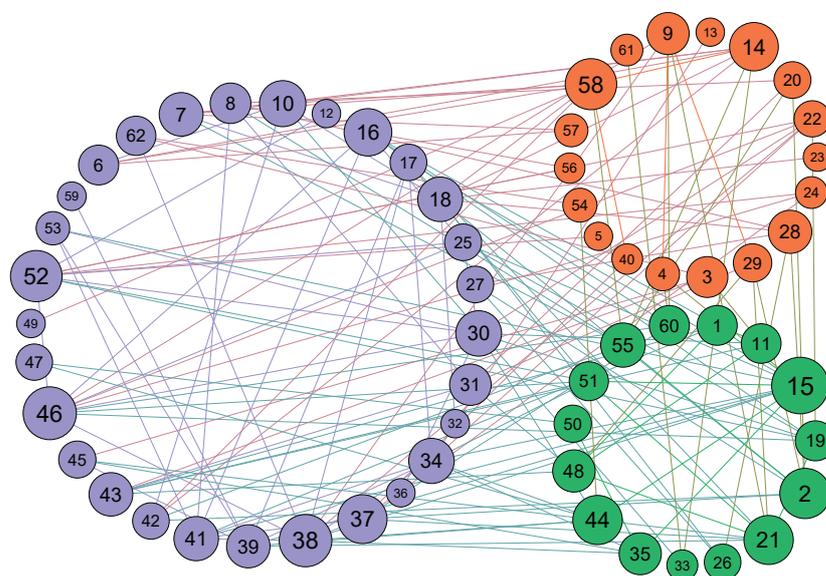


Figure 3. The final community structure of Dolphin according to the NINS algorithm.

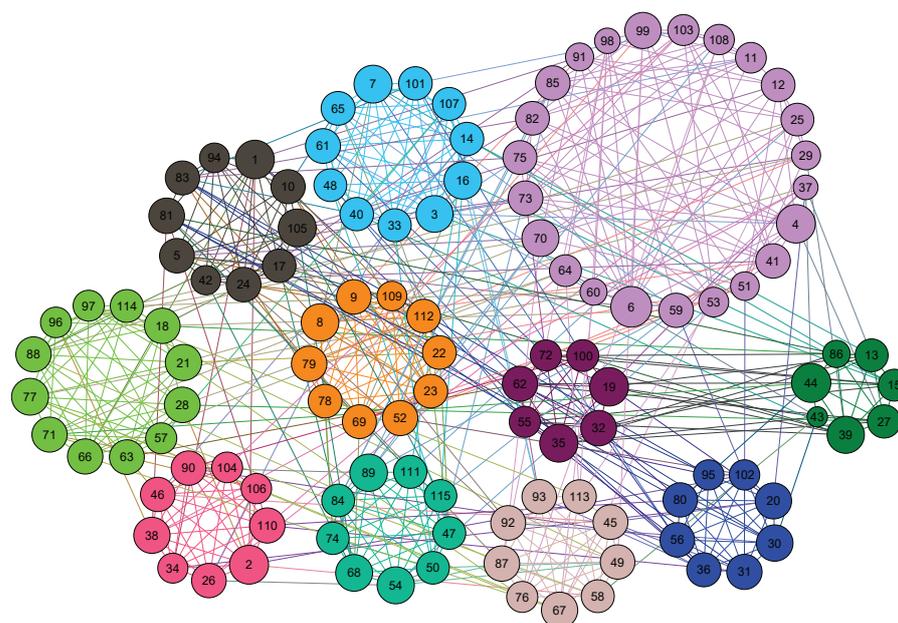


Figure 4. The final community structure of Football according to the NINS algorithm.

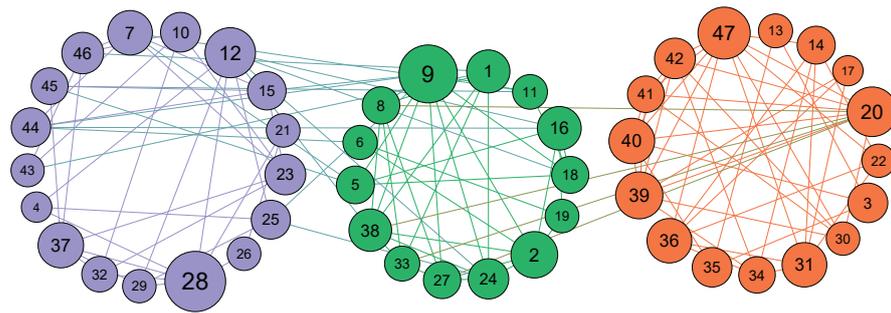


Figure 5. The final community structure of Course registration network according to the NINS algorithm.

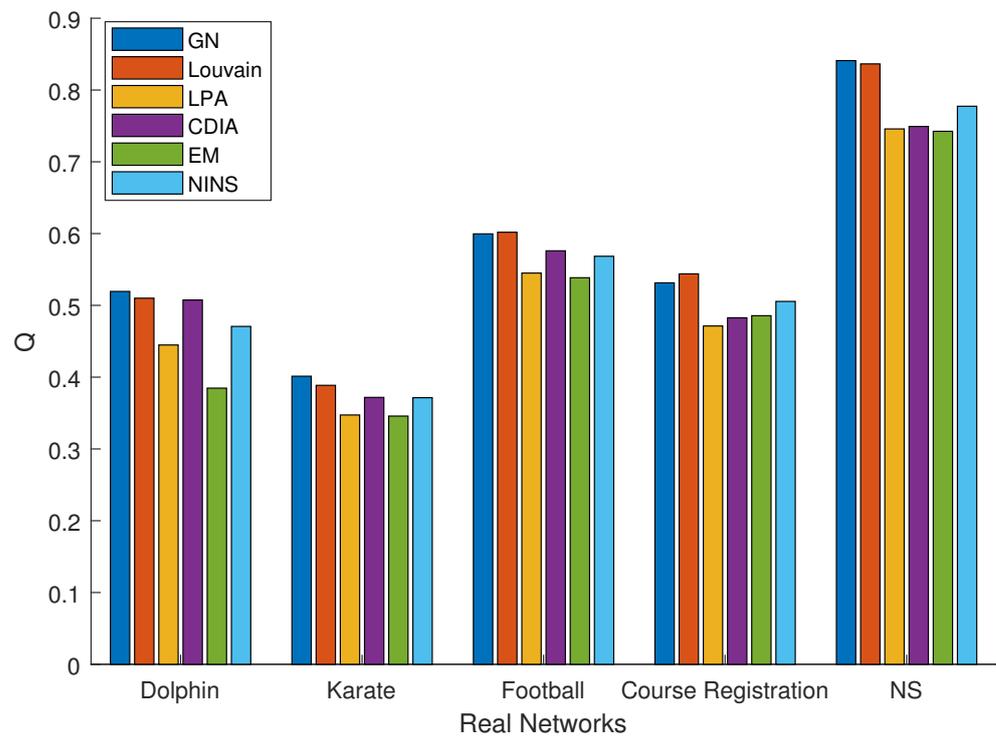


Figure 6. The modularity of NINS and five classic algorithms in five real networks.

Two large networks (Router and Power) were selected to verify the performance of NINS, along with three different algorithms (LPA [13], EM [14], and Louvain [18]). The results of the three classic algorithms are shown in Table 6. As shown in Tables 5 and 6, it is obvious that the modularity of Louvain was higher than that of NINS, but its runtime was higher than that of NINS; the runtime of LPA was lower than that of NINS, but its modularity was lower than that of NINS; the modularity and runtime of EM were the worst. The results obtained in large real networks also indicate that NINS is effective in community detection and it is quite comparable to existing classic methods.

Table 6. The experimental results obtained using three classic methods in large real networks.

Networks	Q(LPA)	Runtime	Q(EM)	Runtime	Q(Louvain)	Runtime
Power	0.5948	0.1753	0.2093	71.8072	0.9319	3.3282
Router	0.3715	0.6967	0.0846	44.14	0.894	2.7594

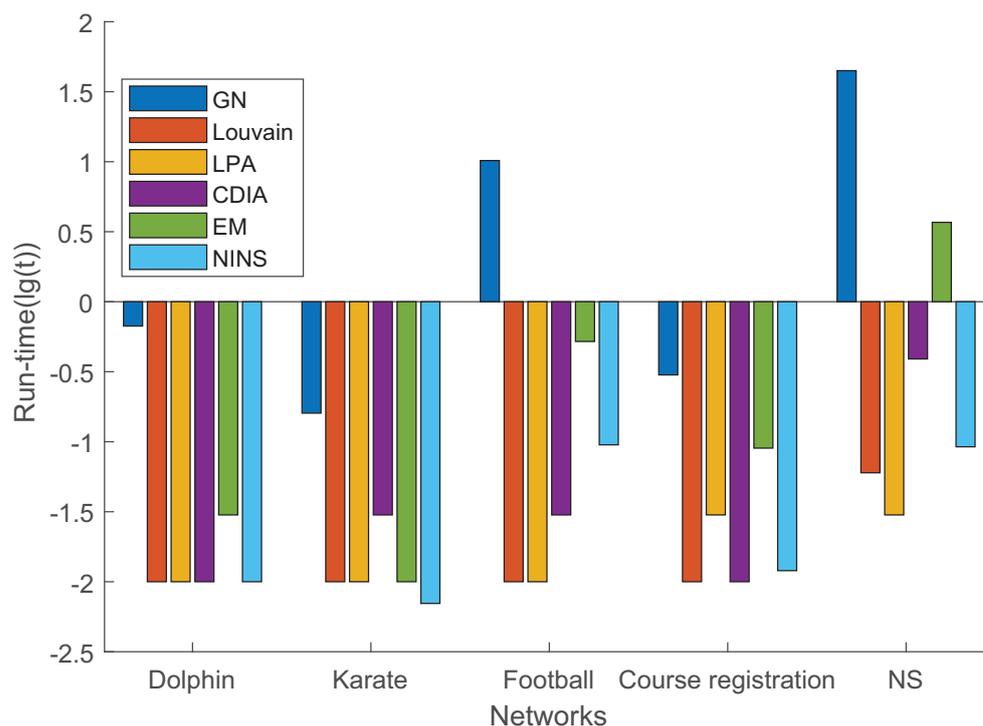


Figure 7. The run-time of NINS and five different algorithms in five real networks.

4.3.2. Experiment on LFR Benchmark Networks

To verify the performance of NINS, comparison experiments were conducted with three classic algorithms (LPA [13], EM [14], and Louvain [18]) in nine LFR networks. Due to the high complexity of GN [12], we did not choose GN as a competitor in this section. The NMI and runtime of four methods in nine LFR networks are shown in Table 7 and Figure 8, respectively. A higher NMI value reflects better performance by the algorithm, which indicates the detection of a more realistic community.

As shown in Table 5, the performance of the proposed NINS was better than that of the other three algorithms when it comes to the detection of realistic communities. As shown in Figure 8, the runtime of the proposed NINS was close to that of the Louvain algorithm with the increase in the number of nodes, which is better than the EM algorithm but worse than the LPA algorithm. The results obtained in LFR networks also indicate that the proposed NINS is effective and quite comparable to existing classic methods.

Table 7. The NMI of four methods in nine LFR networks.

Networks	LPA	Louvain	EM	NINS
LFR-1000	0.99519	0.90853	0.26917	0.99546
LFR-2000	0.99912	0.8901	0.23378	0.99582
LFR-3000	0.99818	0.87236	0.22015	0.99915
LFR-4000	0.99488	0.0.85703	0.20711	1
LFR-5000	0.99701	0.8473	0.20171	1
LFR-6000	0.99664	0.83594	0.19725	0.99973
LFR-7000	0.9975	0.79139	0.189	0.99978
LFR-8000	0.9986	0.83138	0.18918	0.99937
LFR-9000	0.99854	0.83057	0.18648	1

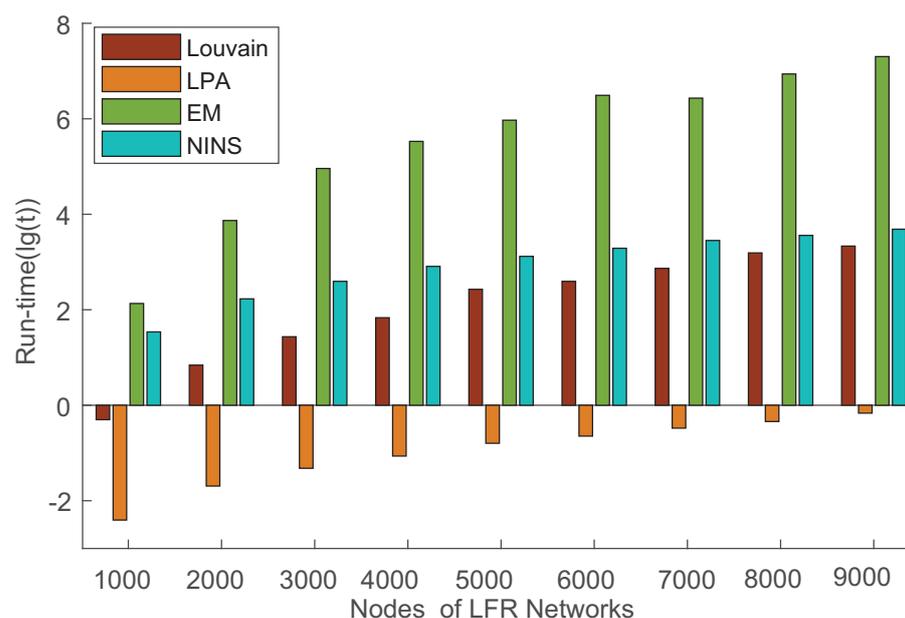


Figure 8. The runtime of NINS and three classic algorithms in LFR networks.

5. Discussion

There are some algorithms that are similar to our method. These algorithms all include identifying the central node, expanding communities, and merging small communities, but the specific methods are different. For example, reference [32] proposes a community detection algorithm that chooses central nodes based on the betweenness and average betweenness, and expands the community by adding the node into the community to which its most similar neighbor belongs. In general, measuring the influence of nodes using betweenness does involve a high time complexity, as well as a low accuracy. The sum of a node's influence on its neighbors better reflects a node's influence. The similarity of connected nodes in the same community is generally larger than the average similarity. Thus, the NINS algorithm proposed in this paper first identifies the central node by calculating the influence of each node, and then expands the community by computing the similarity of nodes and average similarity, and thereafter merges the small communities into the community with the highest similarity. The complexity of NINS is smaller than the algorithm proposed in [32].

Comparing the experiments on the Karate, Dolphin, and Football networks, where the networks represent real communities, the NMI values of the proposed NINS and the algorithm proposed in [32] are (1, 0.2767), (0.603, 0.4699), and (0.8921, 0.8677), respectively; the modularity values of the proposed NINS and the algorithm proposed in [32] are (0.3715, 0.2018), (0.4907, 0.4346), and (0.5684, 0.5857), respectively. The proposed NINS algorithm performs better than the algorithm proposed in [32] in the detection of more realistic and modular communities.

The performance and effectiveness of the proposed algorithm were tested on real and synthetic networks. First, the proposed algorithm was compared with five different algorithms in five small real networks based on modularity and runtime. As shown in Figures 6 and 7, the modularity of the proposed NINS was better than that of LPA [13] and EM [14], close to that of CDIA [19], and lower than that of GN [12] and Louvain [18]. The time complexity of the proposed NINS was better than that of the GN and EM algorithms, and close to that of the Louvain, LPA, and CDIA algorithms. Next, two large real networks were selected to test the performance of NINS. As shown in Table 6, the modularity of NINS was higher than that of LPA and EM but lower than that of Louvain. The runtime of NINS was better than that of EM and Louvain but worse than that of LPA. Finally, the proposed algorithm was compared with three different algorithms in nine LFR networks based on NMI and runtime. As shown in Table 7 and Figure 8, NINS was effective in

detecting more realistic communities when compared with three classic methods (Louvain, EM, and LPA). Although inferior to LPA in terms of time complexity of LPA, it was also quite competitive. In general, the experimental results show that the proposed algorithm is effective in community detection and it is quite comparable to existing classic methods.

6. Conclusions

In this paper, considering node influence and the similarity of nodes, we propose a community detection algorithm named NINS. The proposed algorithm detects non-overlapping communities in unweighted and undirected networks. NINS consists of the following three essential steps: first identifying the central node based on node influence, selecting a candidate neighbor to expand the community based on the similarity of nodes, and then merging the small communities into the community with the most similarity. We compared the proposed algorithm with several classic algorithms in five small real networks and two large real networks, analyzing their modularity and runtime. To show the algorithm's performance and efficiency in networks that have a real community structure, the proposed algorithm was compared with three different algorithms in nine LFR networks, with 1000–9000 nodes, respectively. The results show that NINS performs well in detecting more realistic communities, with a lower computational cost.

Furthermore, during the process of detecting communities, the influential nodes can be obtained, which is a great benefit for better understanding the network. At present, the algorithm detects only non-overlapping communities. It cannot detect overlapping communities. In the future, we will pay attention to the detection of overlapping communities.

Author Contributions: Writing—original draft, Y.X.; Writing—review & editing, T.R. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Joint Fund of Science & Technology Department of Liaoning Province and State Key Laboratory of Robotics, China (2020-KF-12-11), Fundamental Research Funds for the Central Universities (N181706001, N2017009, N2017008, N182608003, N181703005), National Natural Science Foundation of China (61902057).

Data Availability Statement: The datasets are available at <https://www.neusncp.com/api/datasets>, accessed on 19 January 2022.

Acknowledgments: We would like to thank the anonymous reviewers for their careful reading and useful comments that helped us to improve the final version of this paper.

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

References

1. Newman, M.E.J. *Networks*; Oxford University Press: Oxford, UK, 2018.
2. Stam, C. Modern network science of neurological disorders. *Nat. Rev. Neurosci.* **2019**, *15*, 683–695. [[CrossRef](#)]
3. Li, Z.; Ren, T.; Ma, X.; Liu, S.; Zhang, Y.; Zhou, T. Identifying influential spreaders by gravity model. *Sci. Rep.* **2019**, *9*, 8387. [[CrossRef](#)] [[PubMed](#)]
4. Li, Z.; Ren, T.; Xu, Y.; Chang, B.; Chen, D.; Sun, S. Identifying Influential Spreaders Based on Adaptive Weighted Link Model. *IEEE Access* **2020**, *8*, 66068–66073. [[CrossRef](#)]
5. Kamath, P.S.; Wiesner, R.H.; Malinchoc, M.; Kremers, W.; Therneau, T.M.; Kosberg, C.L.; D'Amico, G.; Dickson, E.R.; Kim, W.R. A model to predict survival in patients with end-stage liver disease. *Hepatology* **2001**, *33*, 464–470. [[CrossRef](#)] [[PubMed](#)]
6. Yang, B.; Liu, D.; Liu, J. *Handbook of Social Network Technologies and Applications*; Springer: Boston, MA, USA, 2010.
7. Wang, D.; Li, J.; Xu, K. Sentiment community detection: Exploring sentiments and relationships in social networks. *Electron. Commer. Res.* **2017**, *17*, 103–132. [[CrossRef](#)]
8. Li, C.; Zhang, Y.; Tong, L. A micro-blog personalized recommendation algorithm based on community discovery. *Microelectron. Comput.* **2017**, *34*, 40–44.
9. Moghaddam, A. Detection of Malicious User Communities in Data Networks. Master's Thesis, University of Victoria, Victoria, BC, Canada, 2011.
10. Huang, X.; Chen, D.; Wang, D.; Ren, T. MINE: Identifying Top-k Vital Nodes in Complex Networks via Maximum Influential Neighbors Expansion. *Mathematics* **2020**, *8*, 1449. [[CrossRef](#)]
11. Beni, H.A.; Bouyer, A. TI-SC: Top-k influential nodes selection based on community detection and scoring criteria in social networks. *J. Ambient. Intell. Hum. Comput.* **2020**, *11*, 4889–4908. [[CrossRef](#)]

12. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [[CrossRef](#)] [[PubMed](#)]
13. 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. [[CrossRef](#)] [[PubMed](#)]
14. Newman, M.E.J.; Leicht, E.A. Mixture models and exploratory analysis in networks. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 9564–9569. [[CrossRef](#)] [[PubMed](#)]
15. Kernighan, B.W.; Lin, S. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell Syst. Tech. J.* **1970**, *49*, 291–307. [[CrossRef](#)]
16. Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [[CrossRef](#)]
17. Waltman, L.; Eck, N.J.V. A smart local moving algorithm for large-scale modularity-based community detection. *Eur. Phys. J. B* **2013**, *88*, 471. [[CrossRef](#)]
18. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [[CrossRef](#)]
19. Huang, X.; Chen, D.; Ren, T.; Wang, D. CDIA: A Feasible Community Detection Algorithm Based on Influential Nodes in Complex Networks. In Proceedings of the International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Kunming, China, 20–22 July 2019; Springer: Cham, Switzerland, 2019; pp. 930–937.
20. Palla, G.; Derényi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [[CrossRef](#)] [[PubMed](#)]
21. Chen, D.; Sima, D.F.; Huang, X. Overlapping Community and Node Discovery Algorithm Based on Edge Similarity. In Proceedings of 2017 International Conference on Electronic and Information Technology (ICEIT), Jinan, China, 24–25 November 2017
22. Bedi, P.; Sharma, C. Community detection in social networks. *Wiley Interdiscipl. Rev. Data Min. Knowl. Discov.* **2016**, *6*, 115–135. [[CrossRef](#)]
23. Newman, M.E.J. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *69*, 066133. [[CrossRef](#)] [[PubMed](#)]
24. Arasteh, M.; Alizadeh, S. Community detection in complex networks using a new agglomerative approach. *Turk. J. Electric. Eng. Comput. Sci.* **2019**, *27*, 3356–3367. [[CrossRef](#)]
25. Arasteh, M.; Alizadeh, S. A fast divisive community detection algorithm based on edge degree betweenness centrality. *Appl. Intell.* **2019**, *49*, 689–702. [[CrossRef](#)]
26. Chen, X.; Li, J. Community detection in complex networks using edge-deleting with restrictions. *Phys. A Stat. Mech. Its Appl.* **2019**, *519*, 181–194. [[CrossRef](#)]
27. Basuchowdhuri, P.; Sikdar, S.; Nagarajan, V.; Mishra, K.; Gupta, S.; Majumder, S. Fast detection of community structures using graph traversal in social networks. *Knowl. Inf. Syst.* **2019**, *59*, 1–31. [[CrossRef](#)]
28. Al-Andoli, M.; Cheah, W.P.; Tan, S.C. Deep learning-based community detection in complex networks with network partitioning and reduction of trainable parameters. *J. Ambient. Intell. Hum. Comput.* **2021**, *12*, 2527–2545. [[CrossRef](#)]
29. Agrawal, S.; Patel, A. SAG Cluster: An unsupervised graph clustering based on collaborative similarity for community detection in complex networks. *Phys. A Stat. Mech. Its Appl.* **2021**, *563*, 125459. [[CrossRef](#)]
30. Tsitsekli, K.; Krommyda, M.; Karyotis, V.; Kantere, V.; Papavassiliou, S. Scalable Community Detection for Complex Data Graphs via Hyperbolic Network Embedding and Graph Databases. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1269–1282. [[CrossRef](#)]
31. He, J.; Li, D.; Liu, Y. Modularity-based representation learning for networks. *Chin. Phys. B* **2020**, *29*, 128901. [[CrossRef](#)]
32. Feng, C.; Ye, J.; Hu, J.; Yuan, H.L. Community Detection by Node Betweenness and Similarity in Complex Network. *Complexity* **2021**, *2021*, 9986895. [[CrossRef](#)]
33. Arasteh, M.; Alizadeh, S.; Lee, C.G. Gravity algorithm for the community detection of large-scale network. *J. Ambient. Intell. Hum. Comput.* **2021**, to be published.
34. 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 Syst. Appl.* **2021**, *186*, 115854. [[CrossRef](#)]
35. Riolo, M.A.; Newman, M. Consistency of community structure in complex networks. *Phys. Rev. E* **2020**, *101*, 052306. [[CrossRef](#)] [[PubMed](#)]
36. Cauteruccio, F.; Corradini, E.; Terracina, G.; Ursino, D.; Virgili, L. Investigating Reddit to detect subreddit and author stereotypes and to evaluate author assortativity. *J. Inf. Sci.* **2020**. [[CrossRef](#)]
37. Mengoni, P.; Milani, A.; Poggioni, V.; Li, Y. Community elicitation from co-occurrence of activities. *J. Future Gener. Comput. Syst.* **2020**, *110*, 904–917. [[CrossRef](#)]
38. Naik, D.; Ramesh, D.; Gandomi, A.H.; Gorojanam, N.B. Parallel and distributed paradigms for community detection in social networks: A methodological review. *Expert Syst. Appl.* **2022**, *187*, 115956. [[CrossRef](#)]
39. Yassine, S.; Kadry, S.; Sicilia, M.-A. Detecting communities using social network analysis in online learning environments: Systematic literature review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *12*, e1431. [[CrossRef](#)]
40. Bara'a, A.A.; Abbood, A.D.; Hasan, A.A.; Pizzuti, C.; Al-Ani, M.; Özdemir, S.; Al-Dabbagh, R.D. A Review of Heuristics and Metaheuristics for Community Detection in Complex Networks: Current Usage, Emerging Development and Future Directions. *Swarm Evol. Comput.* **2021**, *63*, 100885.
41. Huang, X.; Chen, D.; Ren, T.; Wang, D. A survey of community detection methods in multilayer networks. *Data Min. Knowl. Disc.* **2021**, *35*, 1–45. [[CrossRef](#)]
42. Calderer, G.; Kuijjer, M.L. Community Detection in Large-Scale Bipartite Biological Networks. *Front. Genet.* **2021**, *12*, 520. [[CrossRef](#)] [[PubMed](#)]

43. Gasparetti, F.; Sansonetti, G.; Micarelli, A. Community detection in social recommender systems: A survey. *Appl. Intell.* **2021**, *51*, 3975–3995. [[CrossRef](#)]
44. Rosvall, M.; Delvenne, J.C.; Lambiotte, R. Different approaches to community detection. In *Advances in Network Clustering and Blockmodeling*; Hoboken, N.J., Ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019; pp. 105–119.
45. Dao, V.; Bothorel, C.; Lenca, P. Community structure: A comparative evaluation of community detection methods. *Netw. Sci.* **2020**, *8*, 1–41. [[CrossRef](#)]
46. Adamic, L.A.; Adar, E. Friends and neighbors on the Web. *Soc. Netw.* **2003**, *25*, 211–230. [[CrossRef](#)]
47. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 36–41. [[CrossRef](#)]
48. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1997**, *33*, 452–473. [[CrossRef](#)]
49. Danon, L.; Diazguilera, A.; Arenas, A. Effect of size heterogeneity on community identification in complex networks. *J. Stat. Mech.* **2006**, *2006*, P11010. [[CrossRef](#)]
50. Lusseau, D.; Schneider, K.; Boisseau, O.J.; Haase, P.; Slooten, E.; Dawson, S.M. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **2003**, *54*, 396–405. [[CrossRef](#)]
51. Lü, L.; Jin, C.; Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Phys. Rev. E* **2009**, *80*, 046122. [[CrossRef](#)] [[PubMed](#)]
52. Watts, D.J.; Strogatz, S.H. Collective dynamics of ‘small-world’ networks. *Nature* **1998**, *393*, 440–442. [[CrossRef](#)] [[PubMed](#)]
53. Spring, N.; Mahajan, R.; Wetherall, D.; Anderson, T. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.* **2004**, *12*, 2–16. [[CrossRef](#)]
54. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [[CrossRef](#)] [[PubMed](#)]