Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.018050

ARTICLE

Seed-Oriented Local Community Detection Based on Influence Spreading

Shenglong Wang[1,*], Jing Yang[1,*], Xiaoyu Ding[2], Jianpei Zhang[1] and Meng Zhao[1]

[1]College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China

[2]College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

[*]Corresponding Authors: Shenglong Wang. Email: wangsl@hrbeu.edu.cn; Jing Yang. Email: yangjing@hrbeu.edu.cn

Received: 25 June 2021 Accepted: 08 December 2021

ABSTRACT

In recent years, local community detection algorithms have developed rapidly because of their nearly linear computing time and the convenience of obtaining the local information of real-world networks. However, there are still some issues that need to be further studied. First, there is no local community detection algorithm dedicated to detecting a seed-oriented local community, that is, the local community with the seed as the core. The second and third issues are that the quality of local communities detected by the previous local community detection algorithms are largely dependent on the position of the seed and predefined parameters, respectively. To solve the existing problems, we propose a seed-oriented local community detection algorithm, named SOLCD, that is based on influence spreading. First, we propose a novel measure of node influence named *k-core* centrality that is based on the *k-core* value of adjacent nodes. Second, we obtain the seed-oriented local community, which is composed of the may-members and the must-member chain of the seed, by detecting the influence scope of the seed. The may-members and the must-members of the seed are determined by judging the influence relationship between the node and the seed. Five state-of-art algorithms are compared to SOLCD on six real-world networks and three groups of artificial networks. The experimental results show that SOLCD can achieve a high-quality seed-oriented local community for various real-world networks and artificial networks with different parameters. In addition, when taking nodes with different influence as seeds, SOLCD can stably obtain high-quality seed-oriented local communities.

KEYWORDS

Complex network; local community detection; influence spreading; seed-oriented; degree centrality; *k-core* centrality; local expansion

1 Introduction

In recent years, complex networks have been prevalent in various domains, such as social media, bioengineering, computer networks and e-commerce shopping [1]. An important property of complex networks is the community structure, which can be defined as a group in which entities are tightly connected [2]. Community structures, in which the entities are represented by the nodes and the relationship between entities are represented by the links, widely exist in the real world [3]. In addition, nodes within the same community are more similar than those between different communities. In other words, a community is in some fashion separated from the other communities [4].

Research on community detection has attracted extensive attention in the last two decades. As an important branch of community detection, local community detection has rapidly developed because of its nearly linear computing time and the convenience of obtaining the local information of real-world networks. Most of the existing local community detection algorithms concentrate on detecting the community with the optimal quality function where a seed is located [5–9]. The community with the optimal quality function is the closest to the community in the real world, and its core members are also the closest to the core members of the real-world community but not necessarily the given nodes. It is also of practical significance to obtain the local members that can be influenced by an individual in the community. Sometimes, we just want to know who has the ability to influence the network, not which community this person is located. In addition, this person may not be important in the community, but there may be others who will be affected by him. Therefore, this paper proposes an algorithm aimed at detecting the local community with the seed as the core for the first time.

Many excellent local community detection algorithms have been proposed. However, some problems still hinder the development of local community detection algorithms. First, as mentioned above, the communities detected by previous local community detection algorithms do not take the seed as the core. This leads to the seed deviation problem. Second, the quality of the detected communities depends on the location of the seed. This leads to the seed dependence problem. Third, the quality of the detected communities depends on the predefined parameter. This leads to the parameter dependence problem. In addition, the predefined parameter makes it difficult and time-consuming to obtain the most reasonable parameter.

This paper proposes a seed-oriented local community detection algorithm, named *SOLCD*, based on influence spreading to solve the three problems mentioned above. In order to solve the seed deviation problem and the seed dependence problem, *SOLCD* expands the community by constantly exploring the influence scope of the seed so the seed is always in the influence center of the detected community. This ensures that the seed is in the core position in the resulting community and the quality of the resulting community. In order to measure the node influence, we propose *k-core* centrality based on the *k-core* decomposition algorithm [10]. In order to solve the parameter dependence problem, *SOLCD* uses the influence spreading method that needs no parameter. In order to verify the quality of the resulting community, we propose a local community effectiveness index (*LCE*) and a local community uniqueness index (*LCU*) to evaluate the quality of the seed-oriented local community. The main contributions of this paper can be summarized as follows:

- We propose a seed-oriented local community detection algorithm for the first time, named *SOLCD*, based on influence spreading. *SOLCD* has the capacity to detect the local communities with seeds as the core, which not only enables people to obtain the seed-oriented local communities, but also makes obtaining the information of the target node more quickly and effectively.

- We propose a new measure of node influence, named *k-core* centrality, based on the *k-core* decomposition algorithm. Empirical evaluations on artificial and real-world networks show that the proposed algorithm based on *k-core* centrality is robust and efficient in detecting seed-oriented local communities.

- We propose two indices that can effectively evaluate the quality of seed-oriented communities: a local community effectiveness index (*LCE*) and a local community uniqueness (*LCU*) index.

- The experimental results on both artificial and real-world networks show that the *SOLCD* has the capacity to detect high quality seed-oriented local communities with stronger robustness than some of the latest algorithms. In addition, *SOLCD* can effectively solve the three problems existing in the research of local community detection: the seed deviation, the seed dependence and the parameter dependence problem.

The rest of this paper is organized as follows. Section 2 reviews the related works on local community detection. In Section 3, we describe the proposed algorithm in detail, and introduce a new node influence measure based on *k-core* centrality. In Section 4, we introduce a local community effectiveness index and a local community uniqueness index to estimate the quality of the detected local communities, then we test the proposed algorithm and compare it with some latest algorithms. Section 5 summarizes our work.

2 Related Works

Most of the existing local community detection algorithms based on the local expansion method consist of two major components: seed selection and community expansion. In the seed selection process, the algorithms select an appropriate node or node set as the seeds to replace the given node so as to be more suitable for community expansion. In the community expansion process, the algorithms expand the community, composed of seeds originally, by running a variety of expansion mechanisms. This section outlines the current research on the local expansion method and describes the dilemma of the current research.

2.1 Seed Selection

Seed selection is widely concerned because of its importance to the local expansion method [11], and various seed selection methods have been proposed. Lancichinetti et al. [12] randomly selected a node as the seed, which makes the results of this algorithm uncertain and leads to a weakness of this algorithm. Similarly, Baumes et al. [13] proposed an algorithm randomly selects edge as the seed. However, in the searching seeds process, this algorithm produces multiple duplicate communities, which consumes considerable time. Lee et al. [14] took a set of k nodes, in which each pair of nodes had an edge, namely, *k-clique*, as the seeds. Whang et al. [15] proposed a new seed selection strategy based on the personal *PageRank* clustering scheme. The key to this algorithm is neighborhood inflation, in which seeds are modified to represent their entire node neighborhood. Ding et al. [16] proposed a robust two-stage local community detection algorithm (*RTLCD*) to detect the core member of the real-world community as a substitute for the given node based on the node relation strength. Cheng et al. [17] scored the nodes in a network using the technique for order of preference by similarity to ideal solution (*TOPSIS*) and took the node with the highest score as the seed. In order to reduce the impact of the seed dependence problem, Guo et al. [18] take the core area which is detected by adding neighboring nodes with the maximum optimized *local modularity density*, as the seeds. Ni et al. [19] took the nodes whose fuzzy relationship with their *NGC* nodes was greater than the threshold as the seeds. An *NGC* node [20] is the nearest node with greater centrality.

These excellent seed selection methods mentioned above can effectively improve the quality of the detected community, but there are still two dilemmas that have not received much attention. First, some seed selection methods, such as the random seed selection method, directly take the given node as the seed [12–15,21]. This leads to the quality of the detected community depending heavily on the location of the seed, which greatly affects the accuracy of the community detection algorithm. Second, some seed selection methods, such as *RTLCD*, *TOPSIS* [16,17,19,20], select the nodes that are more suitable for community expansion as substitutes for the given node, which effectively alleviates the dependence of the algorithm on seed location. However, the replacement of the given node by the seed selection method will cause the given node to deviate from the result community, which makes the existing local community detection algorithms unable to effectively detect the community dominated by the given node.

2.2 Community Expansion

The function of community expansion is to expand the initial community into a local community by adding adjacent nodes to the detected community. Common community expansion methods include the quality function [5–9] and the influence spreading [22–26].

The quality function defines the community structure in a network, which can be used to evaluate the community division quality [27]. Yang et al. [28] studied 13 quality functions and tested their sensitivity, robustness and performance on 230 large real-world networks. Based on this research, Yang et al. [28] classify quality functions into four categories: (1) links within a community, (2) links outside a community, (3) links within and outside a community, and (4) *modularity*.

The main idea of the influence spreading method is to score each node with an influence evaluation mechanism and spread it to the entire network. Raghavan et al. [29] proposed the label propagation algorithm (*LPA*) based on the epidemic spreading model. *LPA* assigns each node in the network a unique label, and then updates the node label to be consistent with the label of its majority neighbors until the label no longer changes. Because of the convenience and efficiency of *LPA*, researchers have successively proposed a series of algorithms based on *LPA*. Xu et al. [30] proposed an improved *LPA* algorithm based on a two-level neighborhood similarity measure named *TNS*, which could help to further divide a network into communities accurately. Inspired by *LPA*, Wu et al. [31] merged communities whose size was smaller than a threshold, where the threshold was based on a reasonable communities' scale, into reasonable communities to increase the community division accuracy. Based on *LPA*, Gregory et al. [32] proposed a method with the ability to detect overlapping communities, named *COPRA* (community overlap propagation algorithm). Tang et al. [33] revealed the overlapping nodes and proposed an algorithm based on the *k-lowest-influence*.

The community expansion methods described above can obtain high-quality local communities, but there are still some problems that need to be addressed vigorously. First, some community expansion methods need to set parameters before their execution [34,35], which make methods difficult and time-consuming to obtain the most reasonable parameter. Second, existing expansion methods are dedicated to expanding the seed to a community which is the most similar to the real-world community. However, in the community expansion process, the given node may be at the edge of the community, or even be removed from the community. That is, there are no expansion methods that focus on the local community with seeds as the core.

3 Motivations and Basic Definitions

3.1 Motivation

As discussed in Section 2, local community detection algorithms have made excellent achievements in terms of the local expansion method, but there are still three problems hindering the development of community detection research: the seed deviation problem, the seed dependence problem and the parameter dependence problem.

The motivation of solving the three problems is as follows. In order to solve the seed deviation problem and the seed dependence problem, we propose a seed-oriented algorithm, which will always take the given node as the seed, and always take the seed as the influence core in the process of community expansion. In order to solve the parameter dependence problem, we propose a local community detection algorithm based on influence spreading without any parameters.

3.1.1 Motivation for Seed-Oriented Local Community

Traditional local community detection algorithms aim to expand from the seed node to the community that is the most similar to the real community. We call these algorithms community quality-oriented local community detection algorithms. However, in the real world, every individual should have the opportunity to build its own local community that takes the individual as the center. For example, different departments with students can be considered as different communities from the perspective of a college. Every student should have the opportunity to build his own local friendship-community consisting of the members from multiple departments. In addition, every student should have the opportunity to become the center of his own local friendship-community. In this paper, we call this type of local community a seed-oriented local community. Different from traditional algorithms, this paper proposes a seed-oriented local community detection algorithm aiming to build the seed-oriented local community of a given node.

The research value of detecting seed-oriented local communities is as follows. First, when the goal is to measure the influence of a person on the other individuals, we only need to detect the seed-oriented local community where this person is located rather than the quality local community, which helps to improve the efficiency of an algorithm. Second, even if an individual is marginalized in a quality oriented community, this individual influences other individuals in a seed-oriented community. In contrast, in a seed-oriented community, the influence of this individual on the other individuals may be greater than that of the core members of a quality oriented community. Third, as a marginal individual in a quality oriented community, the local influence in a seed-oriented community may be greater than that of the core members in the quality-oriented community.

In a community, core members are described as members at the center of the community, hub members are members in close contact with members outside the community, and outlier members are members on the boundary of the community. In fact, core members have higher influence than hub members, and hub members have higher influence than outlier members. Based on this fact, this paper proposes a seed-oriented local community detection algorithm based on influence spreading. The proposed algorithm is guided by the following: a node tends to become a member of the community that is generated by an adjacent node with higher influence.

A sample of the seed-oriented local community detected by the proposed algorithm is shown in Fig. 1. We show a seed-oriented local community C and its neighboring subnetwork N. The figure shows that all the may-members of C have an influence not lower than that of the seed node. In addition, all the must-members of C, which are connected with the seed node using must-member chains, have an influence lower than the seed node.

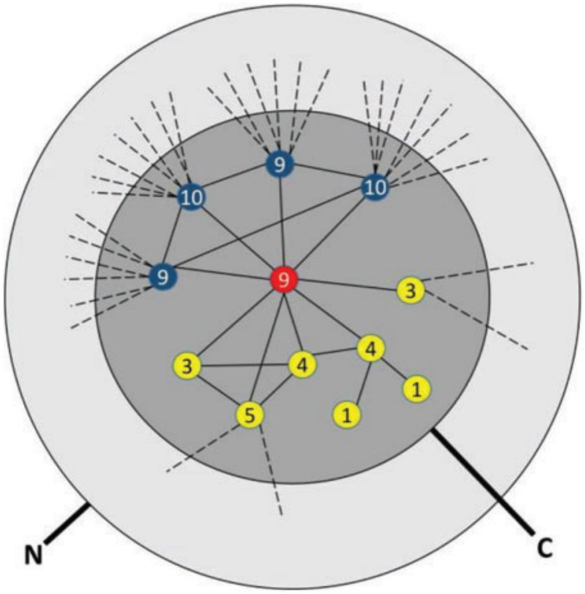


Figure 1: A sample of seed-oriented local community detected by the proposed algorithm. C denotes the seed-oriented local community, N denotes the neighbor sub-network of C. The nodes colored yellow are the must-members of C, the nodes colored blue are the may-members of C, the node colored red is the seed node. The number inside the node is the node influence. Solid lines connect the nodes within C, dotted lines connect the nodes between C and N

3.2 Problem Definition

This paper considers an unweighted graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of links between nodes. The adjacent matrix A is a two-dimensional array which stores the connectivity A_{ij} between nodes in graph G, where $A_{ij} = 1$ denotes node i and node j are connected, otherwise $A_{ij} = 0$. The communities exist in the graph G can be represented as $\mathbf{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{i}\}(\mathrm{C}_1 \cup \mathrm{C}_2, \ldots, \cup \mathrm{C}_\mathrm{i} \subseteq \mathrm{V})$. A community C consists of a set of nodes, where $\mathrm{C} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{j}\}(\mathrm{C} \in \mathbf{C}, \mathrm{v}_\mathrm{i} \in \mathrm{V})$. The seed-oriented local community detection aims to detect a cover C of the graph, $\mathbf{C} = \mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}$ $(\mathrm{C}_1 \cup \mathrm{C}_2 \cup, \ldots, \cup \mathrm{C}_\mathrm{k} \subseteq \mathrm{V})$, where $\forall \mathrm{v}_\mathrm{k} \in \mathrm{V}$, $\exists \mathrm{v}_\mathrm{k} \in \mathrm{C}_\mathrm{k}$ and $\forall \mathrm{v}_\mathrm{j} \in \mathrm{C}_\mathrm{k}$, $\sum_{v_i \in C_k} l_{v_k}^{v_i} \le \sum_{v_i \in C_k} l_{v_j}^{v_i}$ where $\sum_{v_i \in C_k} l_{v_k}^{v_i}$ denotes the sum of length between each node in C_k and v_k.

3.3 Basic Definitions

In this paper, we detect the seed-oriented local communities by detecting the influence scope of the seed in a network. To facilitate readers following along with this paper, we display the research path of this paper in Fig. 2, and the subsection will provide the related definitions of this paper.

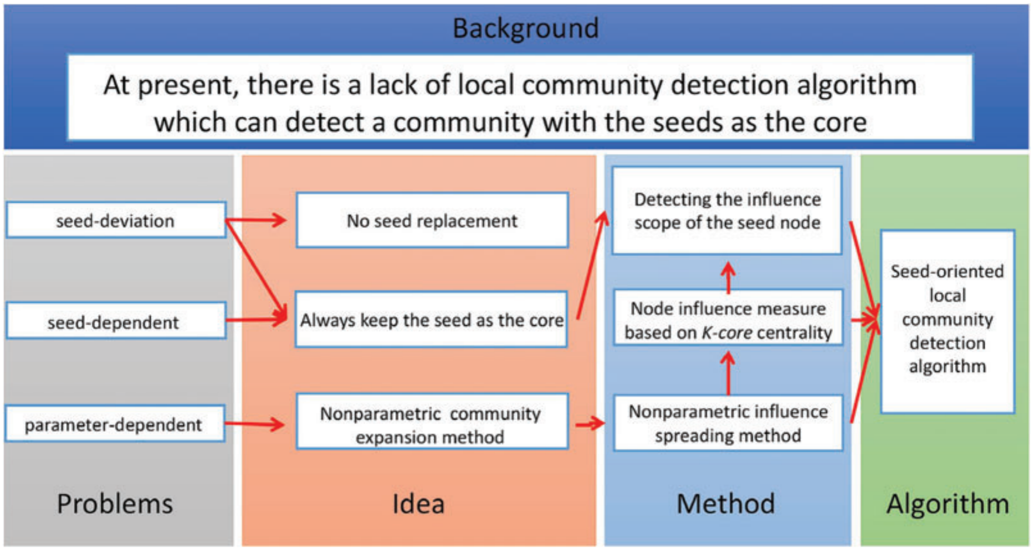

Background
At present, there is a lack of local community detection algorithm which can detect a community with the seeds as the core
seed-deviation
seed-dependent
parameter-dependent
Problems
No seed replacement
Always keep the seed as the core
Nonparametric community expansion method
Idea
Detecting the influence scope of the seed node
Node influence measure based on K-core centrality
Nonparametric influence spreading method
Method
Seed-oriented local community detection algorithm
Algorithm

Figure 2: The research path of this paper

Definition 1 (Node neighbors). The node neighbors of node v are defined as follows:

$$N(v) = \{u | u \in V, A_{uv} = 1\},\ v \in V \tag{1}$$

where A is the adjacent matrix of graph G, and $A_{uv} = 1$ denotes that there is a link between node v and node u.

Definition 2 (Natural community). The natural community of node v is defined as follows:

$$\Gamma(v) = N(v) \cup \{v\},\ v \in V \tag{2}$$

The natural community of node v is a node set composed of node v and its neighbors.

Definition 3 (Must-member). The must-members of node v are defined as follows:

$$Must(v) = \{u | u \in N(v), Inf\,(u) < Inf\,(v)\},\ v \in V \tag{3}$$

The must-members of node v is a node set composed of node v's neighbors that have lower node influence than that of node v.

Definition 4 (May-member). The may-members of node v are defined as follows:

$$May(v) = \{u | u \in N(v), Inf\,(u) \geq Inf\,(v)\},\ v \in V \tag{4}$$

The may-members of node v is a node set composed of node v's neighbors that do not have lower node influence than that of node v.

Property 1 (Transitivity of influence relationship between must-members). Suppose node B is a must-member of node A, and node C is a must-member of node B. Then, the conclusion we can obtain is that there must be a path from node C to node A, that is node C must be reachable to node A, and node A must have no lower influence than node C.

Proof 1. According to the **Definition 3**, we can know that node A is a neighbor of node B and node B is a neighbor of node C, so there must be a path from node A to node B to node C. In addition, node B has lower node influence than that of node A and node C owns lower node influence than that of node B. Therefore, we can conclude that node C must be reachable for node A, and node C must have lower node influence than that of node A.

Definition 5 (Must-member chain). The must-member chain from node A to node B is defined as follows:

$$Mustchain(A,B) = \{v_1, v_2, \ldots, v_i\},\ v_{i+1} \in Must(v_i),\ i \le |V|,\ v_i \in V \tag{5}$$

The must-member chain can be regarded as a queue composed of nodes in the network. The members in the queue are arranged in descending order according to their node influence, and each member in the queue must be a must-member of the previous member.

Definition 6 (Reachable must-member). A reachable must-member of node v is defined as follows:

$$Remust(v) = \{u_1, u_2, \ldots, u_i\},\ \forall sPath(u_i) \subseteq Mustchain(v, u_i),\ i \le |V|,\ u_i \in V,\ v \in V \tag{6}$$

where *sPath(v, u_i)* denotes the nodes on the shortest path from node u to node v.

A reachable must-member of node v is a node which is reachable from node v, and each shortest path from node v to this node must be a must-member chain.

Definition 7 (Seed-oriented local community). The seed-oriented local community of seed node v is defined as follows:

$$SOLCD(v) = \{v\} \cup May(v) \cup Remust(v),\ v \in V \tag{7}$$

The seed-oriented local community of the seed node is a node set composed of the seed, the may-members and all the reachable must-members of the seed node.

3.4 The Proposed Algorithm

In this subsection, we will show the flowchart of the proposed algorithm in Fig. 3, and the pseudocode of the proposed algorithm in Algorithm 1. The proposed algorithm includes two phases: obtaining the may-members phase and obtaining the must-members phase. The processes of each phase are as follows:

Initialization (Line 1). Line 1 initializes list $List_{may}$ and list $List_{must}$ to empty to store the may-members and must-members of seed node respectively. Line 1 assigns the seed v_{seed} to the queue Q.

Obtaining may-members (Lines 2–7). Phase 1 aims to obtain the may-members of the seed node. M in line 4 denotes the node influence evaluation mechanism. Line 3 obtains all the neighbors of the seed node. If the influence of the neighbor v_i is higher than that of the seed node (Line 4), then line 5 assigns v_i to the may-member list $List_{may}$.

Obtaining must-members (Lines 8–26). Phase 2 aims to obtain the must-members of the seed node. When the queue Q is not empty (Line 9), Line 10 removes the first node v_{first} of Q.

Line 11 obtains all the neighbors of v_{first}. If the influence of the neighbor v_i is lower than that of v_{first} and v_i does not belong to the must-member list $List_{must}$ (Line 12), then line 16 sets the flag to true (Line 16). The flag is a Boolean variable that is used to determine whether the node is a must-member of the seed node. Line 14 obtains the node v_n from the union of the neighbors of the seed v_i, Q and $List_{must}$. If the influence of the neighbor v_n is lower than that of v_i (Line 15), then Line 16 sets the flag to false. Lines 15–16 are to ensure that node vi is the reachable must-member of node v_{first}. If the flag is true, Line 20 assigns node *vi* to Q. Line

24 assigns v_{first} to the must-member list $List_{must}$. Line 25 removes v_{first} from Q, and repeats the algorithm until the queue Q is empty.

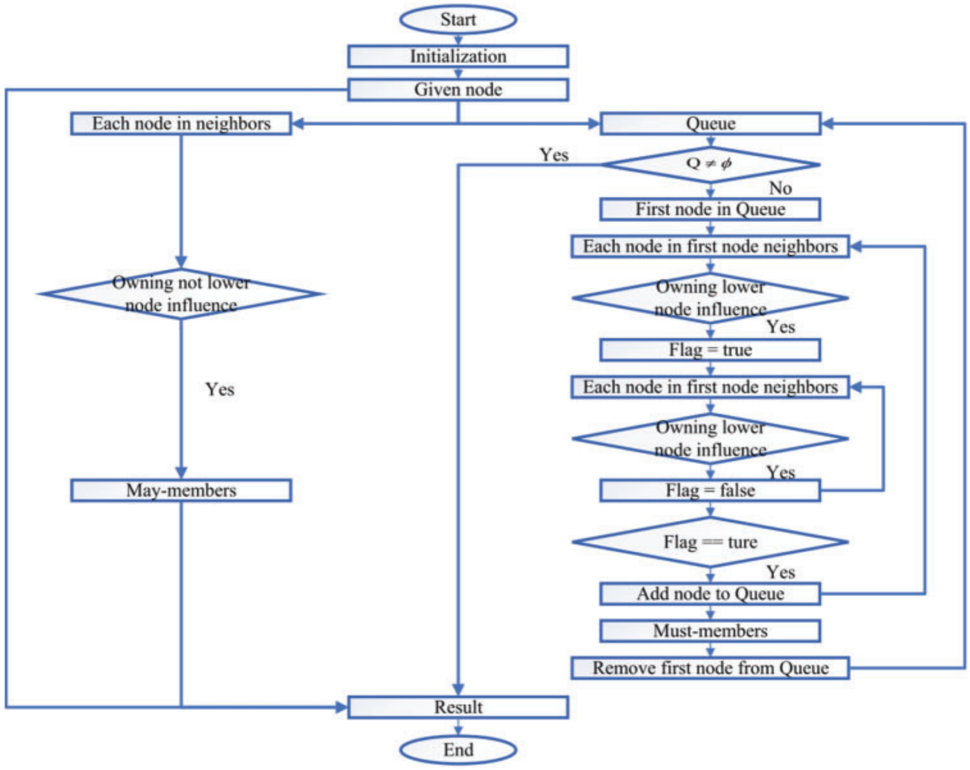


Figure 3: The flowchart of *SOLCD*

Finally, the union of the seed, may-member list $List_{may}$ and must-member list $List_{must}$ is the seed-oriented local community of the seed node v_{seed}.

Algorithm 1: The seed-oriented local community detection algorithm ($SOLCD$)

Input: Graph $G = < V, E >$, link set E, node set V, seed node v_{seed}, Node influence measure Inf.
Output: Community $\boldsymbol{C}$.
Process:

```
1: Initialize $List_{may} = \phi, List_{must} = \phi$, Q = $v_{seed}$
2: Phase 1. Get may-members of $v_{seed}$
3: for all $vi \in N(v_{seed})$ do
4:     if $M_{vi} \geq M_{v_{seed}}$ then
5:         $List_{may} = List_{may} \cup \{vi\}$
6:     end if
7: end for
8: Phase 2. Get must-members of $v_{seed}$
9: while Q $\neq \phi$ do
10:    Get the first element $v_{first}$ of Q
```

(Continued)

Algorithm 1: (Continued)

```
11:   for all $vi \in N(v_{first})$ do
12:     if $Inf_{vi} < Inf_{vfirst}$&&$vi \notin List_{must}$ then
13:       Flag = true.
14:       for all $vn \in N(vi) \cap (Q \cup List_{must})$ do
15:         if $Inf_{vn} < Inf_{vi}$ then
16:           Flag = false
17:         end if
18:       end for
19:       if Flag = true then
20:         $Q = Q \cup \{vi\}$
21:       end if
22:     end if
23:   end for
24:   $List_{must} = List_{must} \cup \{v_{first}\}$
25:   $Q = Q - \{v_{first}\}$
26: end while
27: return $C = \{v_{seed} \cup List_{may} \cup List_{must}\}$
```

3.5 Time Complexity Analysis

The time complexity analysis of the proposed algorithm is on a network G, in where the average degree is $\bar{d}$ and the number of node set is N. In Phase 1, it takes $O(\bar{d})$ to scan the neighbors of the seed node. Phase 2 has three nested iterations: Iteration 1 (Lines 14–18), Iteration 2 (Lines 11–23) and Iteration 3 (Lines 9–26). For Iteration 1, it takes $O(\bar{d} + a)$ to scan the union of Q, the must-member list $List_{must}$ and the neighbors of node v_i, where a is a constant. For Iteration 2, it takes $O(log|N|)$ to add or remove node from Q respectively. So the time complexity of Iteration 2 is $O(max\{\bar{d}^2, \bar{d}log|N|\})$. The time complexity of Iteration 3 is $O(max\{\bar{d}^2, \bar{d}|N|log|N|\})$. In summary, the time complexity of the proposed algorithm is $O(max\{\bar{d}^2, \bar{d}|N|log|N|\})$.

The time complexity of proposed algorithm and comparison algorithms is displayed in Table 1, in where k denotes the mean degree; C denotes the detected community; S denotes the shell sub-network of C; N denotes the neighbor sub-network of C.

Table 1: The time complexity of proposed algorithm and comparison algorithms

Algorithms	Time complexity	References
SOLCD	$O(max\{\bar{d}^2, \bar{d}|N|log|N|\})$	[-]
Clauset	$O(k\bar{d}\overline{|C|}^2)$	[36]
LWP	$O(\bar{d}\overline{|C|}^2)$	[37]
Chen	$O(\bar{d}\overline{|C|}^2|N|)$	[38]
LS	$O(max\{\bar{d}|N||S|, \bar{d}|N|log|N|)$	[39]
LCD	$O(max\{|S|3^{\bar{d}/3}, |S|\overline{|C|}^2)$	[40]
RTLCD	$O(rmax\{\bar{d}\overline{|C|}log\overline{|C|}, \overline{|C|}(\bar{d}log\overline{|C|}) + \overline{d}^4)$	[16]

4 Experiments and Analyses

The experimental environment of this paper is as follows: the proposed algorithm and the comparison algorithms are programmed in *JAVA*; all the programs involved in this paper are running in a computer with *Intel (R) Core (TM) i5-4590 CPU, 3.3 GHz, 16GB RAM*. The experiments are implemented in the proposed algorithm and six comparison algorithms on six real-world networks and three groups of different parameters artificial networks, and the experimental results are verified by four commonly used local community indicators and two proposed by this paper seed-oriented community indicators. Table 2 displays related symbols and their explanations.

Table 2: Symbols and descriptions

Symbols	Descriptions (for network G)
n	The number of nodes
m	The number of links
$\bar{\mathrm{d}}$	The mean degree
$\mathrm{d}_{\mathrm{max}}$	The maximum degree of node
$\overline{\lvert\mathrm{C}\rvert}_{\mathrm{min}}$	The minimum size of the community
$\overline{\lvert\mathrm{C}\rvert}_{\mathrm{max}}$	The maximum size of the community
$\overline{\lvert\mathrm{C}\rvert}$	The average size of the community
μ	The mixing parameter
O_{n}	The number of overlapping nodes
O_{m}	The average number of node overlaps
n_{C}	The number of communities

4.1 *Measures of Node Influence*

K-core centrality. This paper proposes a new measure, the *k-core* centrality, which is based on the *k-core* decomposition algorithm [10], for node influence.

$$K_i = \sum_{j \in N_i} c_j \tag{8}$$

where K_i is the *k-core* centrality of node i, N_i is the neighbors of node i, and c_j is the core value of node j.

K-core. *K-core* [1,41] is a subgraph of network G in which the smallest degree of nodes is k. In *k-core* decomposition algorithms [42–44], the *k-core* is defined as a subgraph of the network where all nodes have a degree not less than k, and a *(k + 1)-core* must be a subgraph of the *k-core*. If we say a node has a core value k, it means that the node belongs to a *k-core* so that the node's core value is the maximum value. In addition, when node A has a higher influence than B, both of the core values and the *k-core* centrality of node A are higher than those of node B.

4.2 *Simple Test of SOLCD*

In order to illustrate the proposed algorithm, we make simple tests on the core members, the hub members and the outlier members of *Karate Club Network* [45]. In the analysis, we use the inner-links to represent the links within a community and the outer-links to represent the links connecting different communities. It is worth noting that the tests do not mean to compare the seed-oriented local communities detected by the proposed algorithm with the real communities in a global sense, but to show the regularities of the distribution of the nodes of seed-oriented local communities. Fig. 4 shows the distribution of *Karate Club Network*, in which node *1* and node *34* represent the administrator and the instructor respectively.

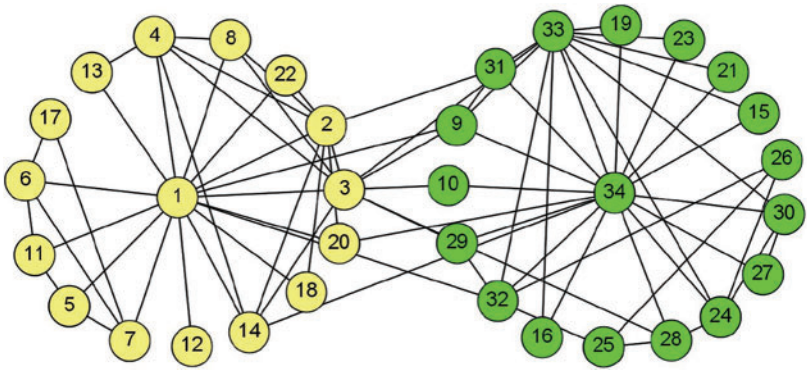


Figure 4: The distribution of karate club network

4.2.1 Tests of the SOLCD on Core Members

From the real network of Karate Club Network, we choose node *1* and node *34* which own the greatest number of inner-links as the core members. From Fig. 5, we can observe that most of the inner-members of the seed-oriented communities are also the inner-members of the real communities.

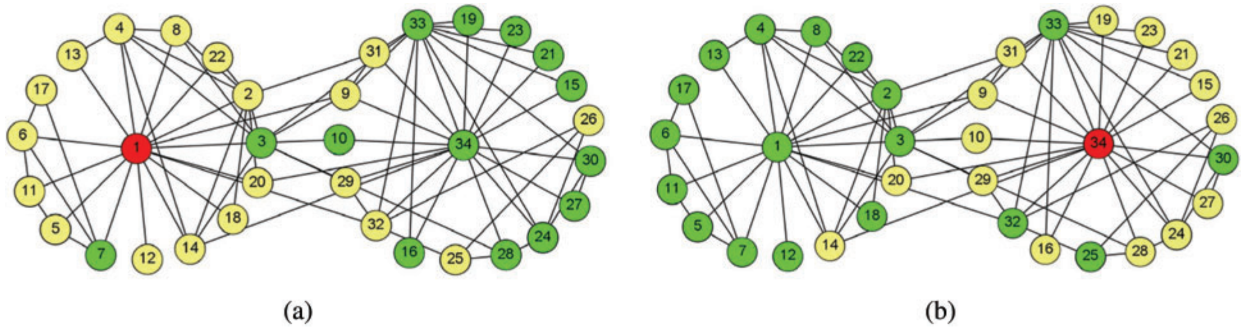


Figure 5: Tests of *SOLCD* on core members. (a) and (b) are the seed-oriented local communities of nodes *1* and *34* generated by *SOLCD*. Nodes colored yellow are the must-members, nodes colored blue are the may-members and nodes colored red are the seed nodes

4.2.2 Tests of the SOLCD on Hub Members

From the real network of Karate Club Network, we choose node *3* and node *9* which have some outer-links as the hub members. From Fig. 6, we can see that the seed-oriented local communities generated from hub members prefer to take the core members (nodes *1* and *34*) as may-members. This phenomenon is because hub members connect different communities and have lower influence than core members. Besides that the communities generated from hub members have smaller size than the communities generated from core members.

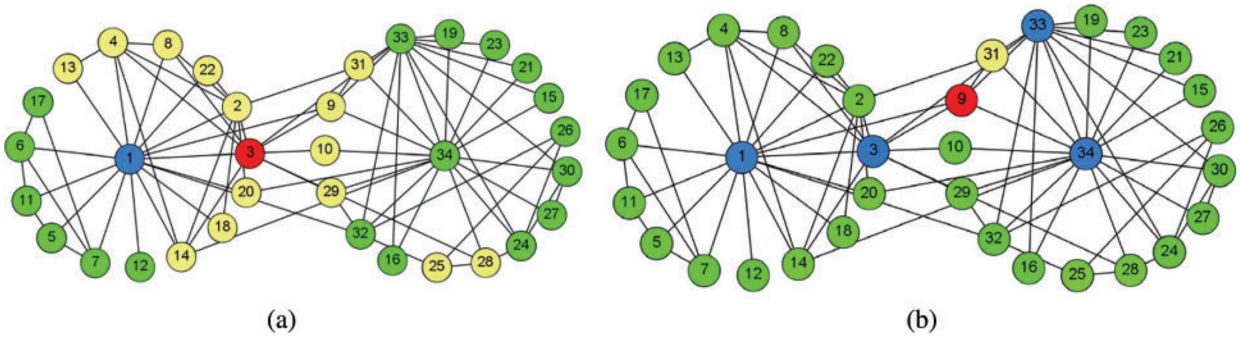


Figure 6: Tests of *SOLCD* on hub members. (a) and (b) are the seed-oriented local communities of nodes *3* and *9* generated by *SOLCD*. Nodes colored yellow are the must-members, nodes colored blue are the may-members and nodes colored red are the seed nodes

4.2.3 Tests of the SOLCD on Outlier Members

From the real network of Karate Club Network, we choose node *12* and node *27* with a few of inner-links as the outlier members. From Fig. 7, we can observe that the seed-oriented local communities generated by the outlier members tend to have more may-members than must-members. This is because outlier members are peripheral members of communities and have lower influence value than that of the core members (nodes *1* and *34*) and that of the hub members (nodes *3* and *9*).

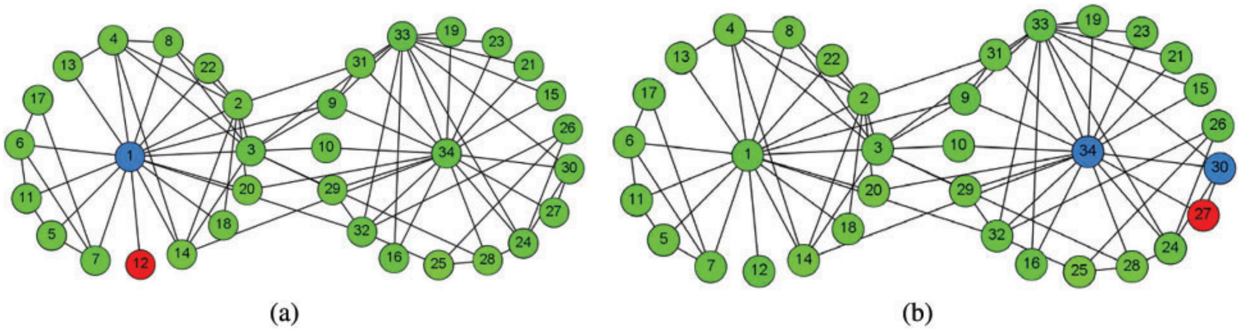


Figure 7: Tests of *SOLCD* on outlier members. (a) and (b) are the seed-oriented local communities of nodes *12* and *27* generated by *SOLCD*. Nodes colored yellow are the must-members, nodes colored blue are the may-members and nodes colored red are the seed nodes

4.2.4 Characteristics of SOLCD

According to samples on core members, hub members and outlier members, we could summarize some characteristics of *SOLCD* as follows:

- *SOLCD* is a self-adaptive algorithm without any help of pre-defined parameters. This avoids the parameter-dependent problem.
- Regardless of the seed node's attributes (core member, hub member or outlier member), the detected seed-oriented local community always take the seed node as core member. This solves the seed-deviation problem.

4.3 *Evaluation Criteria*

Two common used community detection algorithm evaluation criteria are adopted in this paper to verify the performance of *SOLCD*: the Normalized Mutual Information [46] (*NMI*) and *F-score* [47].

4.3.1 *Normalized Mutual Information*

Danon et al. [46] used information entropy as the measurement of the similarity between real-world communities and the resulting communities, which is named normal mutual information (*NMI*). The basis of *NMI* is a confusion matrix N in which the rows represent the information of real-world communities and columns represent the information of the resulting communities. That is, the intersection of real-world communities and resulting communities are represented by element N_{ij} of matrix N, which denotes the numbers of nodes that exist in both communities. *NMI* [46] is defined as follows:

$$\mathrm{NMI}(C_A, C_B) = \frac{-2\sum_{i=1}^{|C_A|}\sum_{j=1}^{|C_B|} N_{ij} log(N_{ij}N/N_{i.}N_{.j})}{\sum_{i=1}^{|C_A|} N_{i.} log(N_{i.}/N)\sum_{j=1}^{|C_B|} N_{.j} log(N_{.j}/N)} \tag{9}$$

where $|C_A|$ represents the number of real-world communities and $|CB|$ represents the number of resulting communities. $Ni.$ and $N.j$ denote the sums of the elements in Row i and Column j, respectively.

NMI is an evaluation index commonly used to assess the community division quality. The better community division quality, the higher the value of *NMI*. The maximum value of *NMI* is *1* when the resulting community is the same as the real-world community.

4.3.2 *F-Score*

F-score [47] is widely used in classification methods to evaluate the quality of the model. The formula of *F-score* is as follows:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

$$Recall = \frac{C_R \cap C_D}{C_G} \tag{11}$$

$$Precision = \frac{C_R \cap C_D}{C_D} \tag{12}$$

where C_R represents the nodes of real-world communities and C_D represents the nodes of detected communities.

Recall is the ratio of the number of correctly found nodes to the number of nodes in the real-world community. *Precision* is the ratio of the number of correctly found nodes to the number of nodes in detected community. *F-score* is the weighted harmonic average of *Recall* and *Precision*.

4.3.3 *LCE*

In this paper, we propose a local community effectiveness index (*LCE*) to measure the quality of seed-oriented local communities. High-quality seed-oriented local communities should satisfy the condition that the seed node is the center of the detected community. In other words, the sum of the shortest path lengths from the seed node to each node of the seed-oriented local community should be smaller than that from other nodes in the rest of the community. *LCE* is defined as follows:

$$LCE_{seed} = 1,\ if\ \forall k \in C_l,\ \sum_{i \in C_l} l^i_{seed} \le \sum_{i \in C_l} l^i_k \tag{13}$$

$$LCE_{seed} = 0,\ if\ \forall k \in C_l,\ \sum_{i \in C_l} l^i_{seed} > \sum_{i \in C_l} l^i_k \tag{14}$$

$$LCE = \frac{\sum_{i \in C_l} LCE_{seed}}{|C_l|} \tag{15}$$

where LCE_{seed} denotes *LCE* value of the seed node seed-oriented local community; C_l denotes the detected local community, and l^i_k denotes the length of the shortest path from node k to node i. We define $LCE = 1$ when the sum of the shortest path lengths from the seed node to the other nodes among all the nodes of the community is the maximum; otherwise, $LCE_{seed} = 0$.

4.3.4 LCU

This paper proposes a local community uniqueness index (*LCU*) to estimate the uniqueness of seed-oriented local communities. A high quality seed-oriented local community should satisfy the condition of having a unique distribution of nodes. *LCU* is defined as follows:

$$LCU = \frac{|\ C_{distintct}\ |}{|\ C_{valid}\ |} \tag{16}$$

where $|C_{distintct}|$ denotes the number of distinct valid local communities, and $|C_{valid}|$ denotes the number of all valid local communities.

4.4 Datasets

4.4.1 Artificial Networks

This paper used Lancichinetti et al. [48] (LFR) benchmark networks to generate various types of artificial networks to evaluate the performance of the proposed algorithm. The *LFR* benchmark network is widely used in the research of complex networks to generate artificial networks that have the same properties as real-world networks. The significance of the parameters affecting the properties of the generated artificial network is as follows. The mixing parameter μ determines the difficulty of detecting the communities for the algorithm. The higher the value of μ is, the harder it is to detect the community structure. $\overline{|\mathrm{C}|}_{\min}$ and $\overline{|\mathrm{C}|}_{\max}$ determine the maximum and minimum size of the communities within the artificial network, respectively; $\bar{\mathrm{d}}$ determines the mean degree of the nodes within the network and d_{max} determines the maximum degree of the nodes within the network; and On and Om determine the overlapping degree of communities in the network. On denotes the number of overlapping nodes between communities and Om denotes the number of overlapping communities of overlapping nodes.

In order to generate different types of artificial networks, we set the parameters of the *LFR* benchmark network as displayed in Table 3, where the expression [a : b : c] represents the value of parameter value ranges from a to c with a spanning of b. As shown in Table 3, we generate artificial networks in three groups of parameters: *LFR*-μ, *LFR*-α_{size} and *LFR*-α_{degree}. These three parameters are used to test the performance of the proposed algorithm in community structure identification, community diversity and node diversity. In order to avoid the influence of the randomness of the generated artificial networks, we generate 10 artificial networks for each parameter and take the average value as the experimental results.

Table 3: The parameter configuration for *LFR* benchmark network

Network	n	$\bar{d}$	d_{max}	$\overline{\|C\|}_{min}$	$\overline{\|C\|}_{max}$	μ	O_n	O_m
LFR-μ	1000	5	25	10	50	[0.1:0.1:0.8]	0	0
LFR-α_{size}	1000	5	25	5 × [1:1:5]	50 × [1:1:5]	0.2	0	0
LFR-α_{degree}	1000	[5:1:15]	5 × [5:1:15]	10	50	0.2	0	0

4.4.2 Real-World Networks

This paper used *6* real-world networks to test the performance of the proposed algorithm. The characteristics of the real-world networks are listed in Table 4. By observing the relationship between *34* members of a karate club at an American university, Zachary et al. proposed the *karate club network* [45] in which nodes represent the members of the club and the links between nodes represent the relationships between nodes. By observing the habits of 62 bottlenose dolphins living in New Zealand, Lusseau et al. [49] found that the communication of these dolphins showed a specific pattern and proposed the *dolphin network*, in which each node represents a bottlenose dolphin and the link between two dolphins represents that these two dolphins are in frequent contact. The *books network* is a network of the purchasing records of political books on Amazon [50]. In the network, the nodes represent political books and a link between two books indicates that they are purchased together frequently. The *football network* is the records among the college teams that participated in the 2000 American football season [51]. In the network, each node represents a participating university and a link means that there was a match between two colleges. The *Amazon network* is a network of purchasing records on Amazon [28]. The *DBLP* network is a network of a scientific collaboration network where nodes denote authors and edges denote that the connected authors have corporations [28]. In addition, in order to obtain more detailed experimental results, we divide *DBLP* into *11* subnetworks according to the community size. The characteristics of *DBLP* after processing are displayed in Table 5.

Table 4: The characteristics of real-world networks

Network	n	$\bar{d}$	d_{max}	$\overline{\|C\|}_{min}$	$\overline{\|C\|}_{max}$	μ	O_n	O_m	Reference
Karate	34	156	4.58	2	17.00	0.128	0	—	[45]
Dolphin	62	318	5.12	2	31.00	0.038	0	—	[51]
Football	115	1226	10.66	12	9.58	0.357	0	—	[50]
Books	105	440	8.38	3	35.0	0.159	0	—	[49]
Amazon	16716	97478	5.83	1163	15.16	0.005	867	2.06	[28]

4.5 Experimental Settings

We compared *SOLCD* to *6* state-of-the-art local community detection algorithms: *RTLCD* (a robust two-stage local community detection algorithm) [16], Clauset et al. [36], LWP (Luo, Wang and Promislow) [37], Chen et al. [38], *LS* (*link similarity*) [39] and *LCD* (*local community detection based on maximum cliques*) [40].

Table 5: The characteristics of DBLP sub-networks

Network (ID)	|C|	n	m	$\bar{d}$	n_c	$\overline{\|C\|}$	μ	O_n	O_m
DBLP(1)	(0, 10]	24210	65650	5.42	3532	7.28	0.107	1311	2.11
DBLP(2)	(10, 20]	14540	91872	6.31	1100	13.85	0.088	632	2.10
DBLP(3)	(20, 30]	3240	21708	6.70	136	24.13	0.029	39	2.05
DBLP(4)	(30, 40]	1338	10352	7.73	39	34.31	0.001	0	—
DBLP(5)	(40, 50]	611	4204	6.88	14	43.64	0.000	0	—
DBLP(6)	(50, 100]	583	3940	6.75	9	64.78	0.001	0	—
DBLP(7)	(100, 200]	1492	9670	6.48	10	150.5	0.029	13	2.0
DBLP(8)	(200, 300]	1341	6618	4.93	6	232.83	0.046	56	2.0
DBLP(9)	(300, 400]	2133	9964	4.67	6	355.67	0.005	1	2.0
DBLP(10)	(400, 500]	834	5670	6.79	2	417.0	0.004	0	—
DBLP(11)	(500, 1000]	10705	70208	6.55	14	785.5	0.086	285	2.02

The *RTLCD* algorithm is a robust two-stage local community detection algorithm that detects the core member of the target community to replace the seed node in the seed selection stage and expands the community based on the relation strength in the community expansion stage [16]. The *Clauset* algorithm extends the *modularity* [36] to the local community, and expands the community by adding nodes that optimize the local community *modularity* ΔR [36]. The *LWP* algorithm improves the local community *modularity* to the *indegree* divided by the *outdegree* and adds the termination condition of the algorithm [37]. The *Chen* algorithm proposes a metric $L = L_{in}/L_{ex}$ which is the internal relation divided by the external relation [38].

Based on the definition of *NMI* and *F-score*, the detected local community has a high value of *NMI* and *F-score* is similar to the real community in a global sense. However, the goal of seed-oriented local community detection is to detect a local community with the seed node as the core member. In fact, some real-world networks have shown the power law distribution of the node degree and the core member occupies only a small scale of the networks. Therefore, in the local communities with a high *NMI* and *F-score* means the seed node cannot become the core member in most cases which indicates that the seed-deviation problem occurs.

Based on the definitions of *Precision* and *Recall*, in a detected community with high *Precision* and low *Recall* means most of the members of this detected community are also the members of the real-world community in a global sense. It is common sense that most of the members of a local community should be a subset of a global community. Therefore, high *Precision* and low *Recall* means that an algorithm prefers to detect communities in a local sense rather than detect communities in a global sense.

When the communities detected by the algorithm have high *Precision* and low *Recall*, which means that the algorithm is more inclined to detect communities in the local sense rather than in the global sense. When the detected communities have high precision and low recall, which means that the algorithm is more inclined to detect communities in the local sense rather than in the global sense. Based on the definition of *LCE*, the community results detected by an algorithm have high *LCE*, which means that the unique local communities detected by the algorithm occupy a higher proportion. Note that, we define a "seed-oriented" local community as a local community in which the seed node must satisfy LCE_{seed}=1 and have high *LCE*.

The experiments are conducted on *6* real-world networks and 3 groups of *LFR* artificial networks. Note that an algorithm running more than 24 h on a single dataset will be terminated.

4.6 Experimental Results on Real-World Networks

Table 6 lists the *NMI*, *Recall*, *Precision*, *F-score*, *LCE*, *LCU* and *Time* metrics of the proposed algorithms and the other comparison algorithms on five real-world networks.

Table 6: The characteristics of the DBLP sub-networks

Network	Criteria	SOLCD	RTLCD	Clauset	LWP	Chen	LS	LCD
Karate	NMI	0.1658	**1.0000**	0.2992	0.5160	0.1552	0.1688	0.4093
	Recall	0.3117	**1.0000**	0.5527	0.6912	0.2071	0.2339	0.6182
	Precision	0.9172	**1.0000**	0.9088	0.8019	0.6345	0.5588	0.8449
	F-*S*core	0.4261	**1.0000**	0.6474	0.7179	0.2949	0.3171	0.6918
	LCE	**0.97**	0.06	0.21	0.12	0.50	0.21	0.07
	LCU	**1.00**	0.06	0.21	0.18	0.29	0.21	0.15
	Time (ms)	0	2	1	1	1	0	1
Dolphin	NMI	0.1191	**0.4526**	0.1857	0.2809	0.0959	0.0709	0.2616
	Recall	0.2217	**0.6399**	0.3013	0.3696	0.1517	0.0980	0.3853
	Precision	0.9346	0.9647	0.9694	0.5271	0.7043	0.4032	**0.9546**
	F-*S*core	0.3351	**0.7376**	0.4287	0.4173	0.2364	0.1458	0.5086
	LCE	**0.92**	0.06	0.35	0.08	0.35	0.21	0.12
	LCU	**1.00**	0.06	0.34	0.21	0.34	0.18	0.16
	Time (ms)	0	1	1	1	2	0	2
Football	NMI	0.4107	0.5146	0.5712	**0.6023**	0.5863	0.5714	0.5638
	Recall	0.7660	**0.9209**	0.7133	0.6409	0.6665	0.5956	0.7280
	Precision	0.5909	0.5568	0.6466	0.6257	**0.6456**	0.6461	0.6354
	F-*S*core	0.6534	0.6639	**0.6689**	0.6301	0.6479	0.6180	0.6708
	LCE	**1.00**	0.31	0.56	0.56	0.56	0.54	0.22
	LCU	**1.00**	0.06	0.50	0.10	0.23	0.13	0.16
	Time (ms)	0	6	3	0	5	0	3
Books	NMI	0.1110	**0.4881**	0.2687	0.2925	0.0905	0.0106	0.3594
	Recall	0.2354	**0.8681**	0.4387	0.4710	0.1532	0.0195	0.6032
	Precision	**0.8162**	0.7049	0.7656	0.4643	0.5720	0.1705	0.7554
	F-*S*core	0.3234	0.7640	0.4982	0.4619	0.2219	0.0307	0.6210
	LCE	**0.91**	0.02	0.18	0.05	0.27	0.13	0.06
	LCU	**1.00**	0.02	0.34	0.09	0.30	0.10	0.09
	Time (ms)	0	5	8	2	5	0	25
Amazon	NMI	0.4529	**0.7254**	0.5668	0.6261	0.4235	0.3918	0.6888
	Recall	0.3954	**0.6966**	0.5192	0.5977	0.3772	0.3641	0.6551
	Precision	**0.9967**	0.9914	0.9964	0.8783	0.8307	0.6776	0.9958
	F-*S*core	0.4980	**0.7570**	0.6138	0.6531	0.4638	0.4122	0.7213
	LCE	**0.91**	0.18	0.34	0.23	0.36	0.31	0.13
	LCU	**0.89**	0.16	0.28	0.14	0.27	0.18	0.11
	Time (ms)	2	0	1	0	1	0	4

As shown in Table 6, we can observe that *SOLCD* achieves the highest *LCE* and *LCU* on all the real-world networks, and the precision is also excellent among all algorithms, especially on *books* and *Amazon*. Although *NMI*, *Recall* and *F-score* of these three indicators achieved by *SOLCD* are not good enough, we know from the analysis in Section 4.5 that high-quality seed-oriented local communities are mainly measured by *LCE, LCU* and *precision* of three indicators rather than *NMI*, *Recall* and *F-score*. Therefore, *SOLCD* can achieve high-quality seed-oriented local communities among real-world networks. *RTLCD* is excellent in *NMI* and *F-Score* on all the real-world networks, which illustrates that *RTLCD* is an outstanding community quality-oriented local community detection algorithm. However, *RTLCD* obtains the worst on *LCE* and *LCU*, which proves that *RTLCD* is severely affected by the seed deviation problem. For the remaining comparison algorithms, *Clauset*, *Chen, LWP*, *LS* and *LCD,* the performance on various indicators is mediocre. That is, these algorithms have a certain ability to detect the seed-oriented local community and community quality-oriented community, but they are not skilled at this.

As shown in Figs. 8a, 8d and 8f, the performance of all the algorithms worsens as the *ID* of *DBLP* increases. As Table 6 shows, the average size of the community increases as the dataset ID increases, which is the main factor that can affect the results of algorithms. The reason for this is that the increase of community size makes the edge of community become more loose, which makes algorithms more difficult to detect the community structure.

Figs. 8b, 8c and 8e show that *SOLCD* is stable and achieves the highest *LCE*, *LCU* and *Precision*, which proves that *SOLCD* has the ability to detect high-quality seed-oriented local communities. *RTLCD* is excellent in *NMI* and *F-score*, which illustrates that *RTLCD* is skilled at detecting local communities in the global sense. However, *RTLCD* obtains the lowest *LCE*, which indicates that it has a serious seed-deviation problem. *Chen* and *LS* have good *LCE* performance, but they also have a seed-deviation problem to a certain extent.

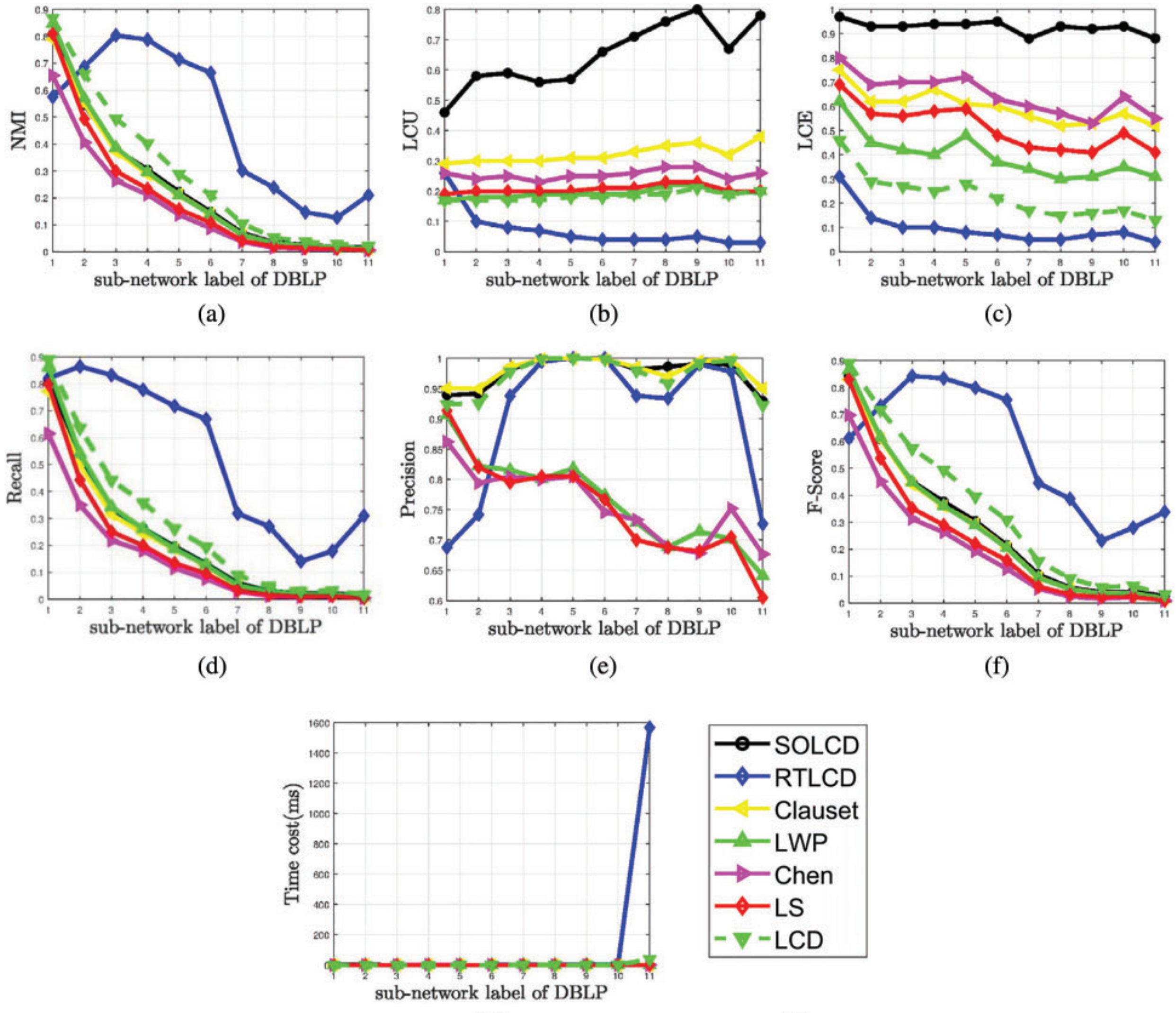
The experimental results on real-world networks show that *SOLCD* has a great ability to achieve high-quality seed-oriented local communities among real-world networks, which proves that *SOLCD* solves the seed-deviation problem. *RTLCD* can achieve the communities with the best community quality, but it is poor at detecting seed-oriented local communities. The rest of the comparison algorithms are more or less affected by the seed deviation problem.

4.7 Experimental Results on Artificial Networks

4.7.1 Experimental Results on LFR-μ

LFR-μ aims to verify the ability of algorithms to reveal the community structure in response to changes in the difficulty of revealing the community structure. Fig. 9 shows the performance of all the algorithms on the *LFR-μ* artificial networks. We observe that all the algorithms show a downward trend on the metrics of *NMI*, *Recall*, *Precision* and *F-score*. This phenomenon occurs because the community structure becomes increasingly more difficult to find as the mixed parameter μ increases.

Figs. 9a and 9f show that *LCD* and *RTLCD* perform excellently in *NMI* and *F-score*, which shows that these algorithms are good at detecting local communities in the global sense. However, as shown in Figs. 9d and 9e, *LCD* and *RTLCD* have high *Recalls* and low *Precisions*, which illustrate that these two algorithms have serious seed deviation problem. Fig. 9b in which *LCD* performs the worst in *LCE*, confirms this matter. In contrary, *SOLCD* and *Chen* have low *Recalls* and high *Precisions*, which indicate that although these two algorithms find only a small number of neighbors of the seed node, these neighbors are the correct members of the seed-oriented community. Fig. 9b affirms this statement. *LCE* of *Chen* is obviously higher than those of other algorithms and *SOLCD* achieves the optimal *LCE* value. As shown in Fig. 9c, *SOLCD* achieves the highest *LCU*.



(a) (b) (c)

(d) (e) (f)

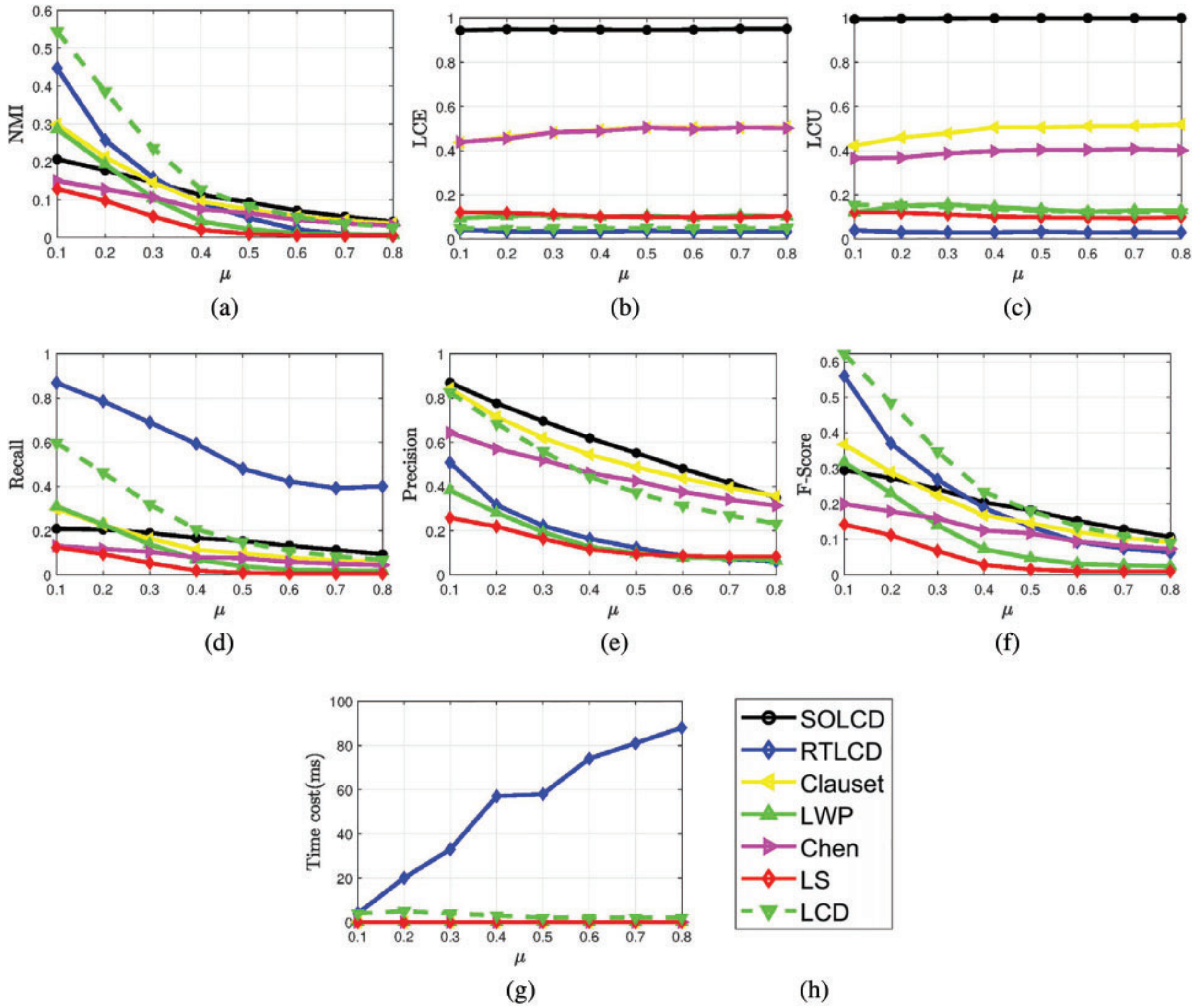
(g) (h)

Figure 8: (a–h) The performance of algorithms on DBLP

The experimental results show that *SOLCD* can achieve high-quality seed-oriented local communities as the mixed parameter μ changes, which solves the seed deviation problem.

4.7.2 Experimental Results on LFR-α_{size}

LFR-α_{size} aims to verify the ability of algorithms to reveal the community structure when the community size changes. Fig. 10 shows the performance of all the algorithms on the *LFR-*α_{size} artificial networks. The scale values of the *x-axis* at the top and bottom of the graph represent the maximum and minimum of community sizes, respectively. As shown in Fig. 10, the results of most of the algorithms on the *NMI*, *Recall*, *Precision* and *F-score* worsen as the maximum and minimum community size increase. The reason is as follows. As the maximum and minimum community size increase, the community structure becomes more diverse, and the boundary of the community becomes fuzzy, which makes it difficult for the algorithms to identify the community structure. Fig. 10c shows that *SOLCD* is stable at the highest *LCU*.



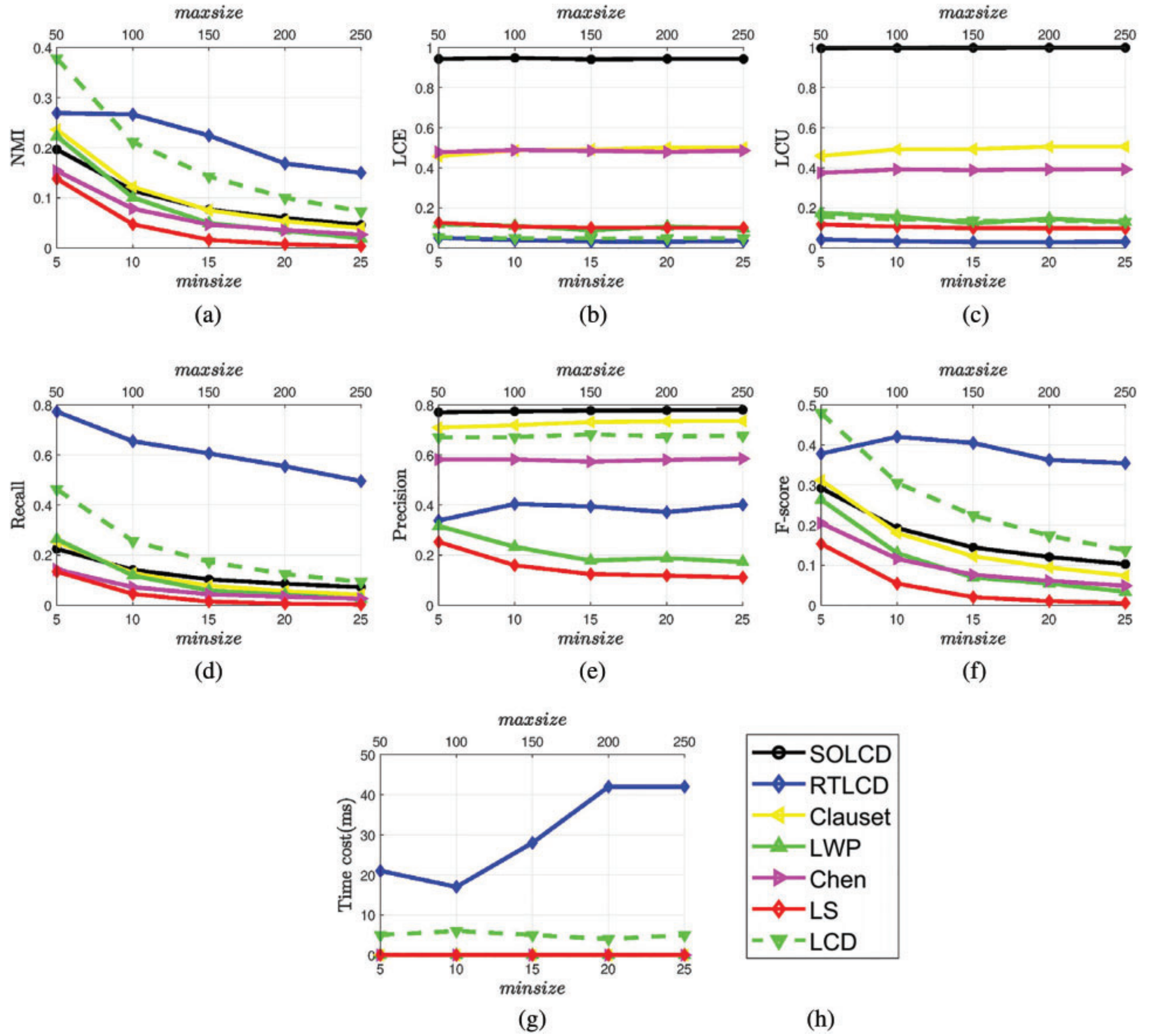
(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 9: (a–h) The performance of algorithms on *LFR*-μ

As shown in the Figs. 10b and 10f, *SOLCD* is stable at a high level on *LCE*, *LCU* and *Precision* regardless of whether the parameter α_{size} changes, which proves that SOLCD can effectively detect seed-oriented local communities. *RTLCD* and *LCD* are stable on the indices of *NMI* and *F-score*, which illustrates that these two algorithms are robust to the parameter α_{size} in detecting local communities in the global sense. However, *RTLCD* and *LCD* perform extremely poorly on *LCE* and *LCU*, which indicates that these methods have serious seed deviation problems. *Chen* and *Clauset* have good performance on the index of Precision, and moderate performance on *LCE* and *LCU*, which shows that these two algorithms have certain capabilities to detect seed-oriented local communities, but they still have seed deviation problems to a certain extent.



(a) (b) (c)

(d) (e) (f)

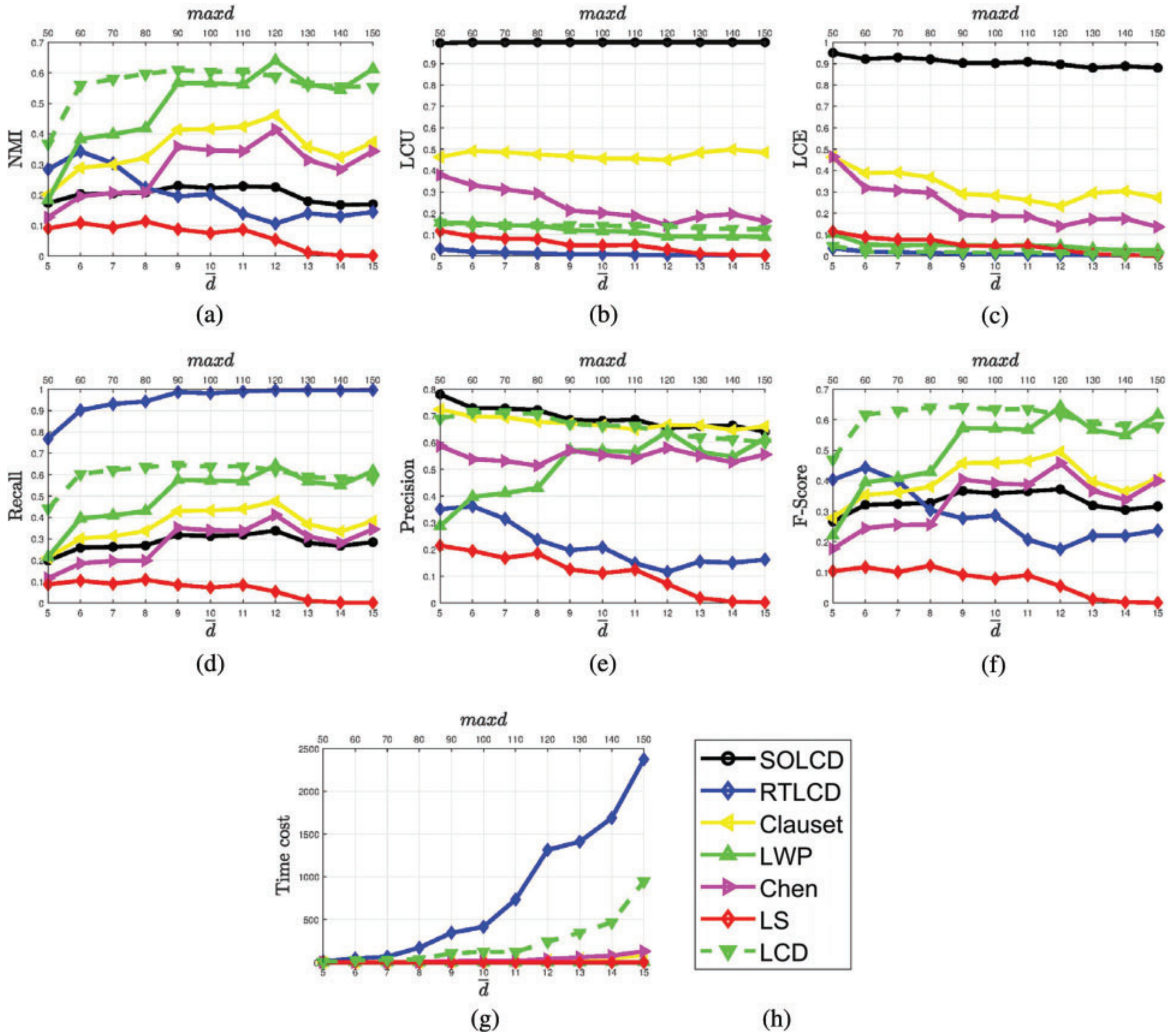
(g) (h)

Figure 10: (a–h) The performance of algorithms on *LFR*-α_{size}

The experiments prove that *SOLCD* is robust to changes in the community size. As the the maximum and minimum community size increase, *SOLCD* can still achieve high-quality seed-oriented local communities which indicates that *SOLCD* solves the seed deviation problem.

4.7.3 Experimental Results on LFR-α_{degree}

LFR-α_{degree} aims to test the performance of algorithms on revealing the community structure as the node degree changes. Fig. 11 displays the results of all the algorithms on the *LFR*-α_{degree} artificial networks. The scale values of the x-axis at the top and bottom of the graph represent the maximum and mean degree of nodes in the network respectively. Figs. 11a and 11f show that the performance of most algorithms on *NMI* and *Precision* improve slightly as the parameter α_{degree} increases. The reason for this outcome is as follows. Increasing the parameter α_{degree} makes the relationship between nodes become more diverse, so it can provide more node information which makes the algorithms easier to detect the community structure; however, it also increases the complexity of the network, which prevents algorithms from exploring the community structure. Therefore, the curves fluctuate.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 11: (a–h) The performance of algorithms on *LFR*-α_{degree}

Fig. 11 shows that *LCE* of *SOLCD* decreases slightly as the parameter α_{degree} increases, but it remains at a high level. The *LCU* and *Precision* of *SOLCD* are outstanding. *LCD* and *LWP* perform excellently on the indices of *NMI* and *F-score*, which indicates that these two algorithms have great abilities to detect the local communities in the global sense. Unfortunately, *LCD* and *LWP* obtain poor *LCEs* which proves that they experience the seed deviation problem. *Clauset* and *Chen* have good performance on *Precision*, and low *LCUs* and *LCEs*, which proves that these two algorithms have certain seed-oriented local community detection abilities but they are not considerable. *RTLCD* underperforms in all indicators except *Recall*, which illustrates that *RTLCD* can neither effectively detect local community in global sense nor detect seed-oriented communities in a local sense when the network has a high degree.

The results indicate that SOLCD has some seed-deviation problems as the mean and maximum node degree increase, but it can achieve a high-quality seed-oriented local community.

4.7.4 Experimental Results for the Seed Dependence Problem

To perform a detailed analysis of the *seed dependence* problem, this paper lists the valid communities generated by seed nodes with different node influences. We take the degree centrality as the node influence measure. We divide all the nodes into ten parts according to their node influence in descending order. Taking Fig. 12 as an example, Fig. 12a is the distribution of the valid seed-oriented local communities detected by the algorithms on the *DBLP1* network, and so on. The abscissa represents the ranking of the seed's node influence among the node influences of all nodes in the network (e.g., '0.1' represents that the seed's node influence is in the top 10% of all nodes, and '1' represents that the bottom 10% of seed nodes). The ordinate represents the proportion of valid seed-oriented local communities in all communities detected by the algorithms. Table 7 displays the standard deviation (*SD*), arithmetic mean and coefficient of variation (*CV*) of the proportion of valid seed-oriented local communities detected by all the comparison algorithms on a group of *DBLP* networks. The **standard deviation** is a measure of the dispersion of the data distribution, which is used to measure the deviation of data from the arithmetic mean. The smaller the standard deviation is, the less these values deviate from the mean, and vice versa. When comparing the dispersion of the two groups of data, the measurement scales of the two groups of data are too different to be compared directly using the standard deviation. At this time, the **coefficient of variation** is required, which is the ratio of the standard deviation and arithmetic mean.

Fig. 12 shows that the curve of *SOLCD* is stable at the top of Figs. 12a–12f but fluctuates in Figs. 12g–12k. The above figures show that the nodes in the middle ranking have more difficulty obtaining local communities with them as the core than the nodes at the top of the ranking and the nodes at the bottom of the ranking. There are three reasons for this phenomenon. First, the nodes at the bottom of the ranking have a small influence scope, so it is easy to obtain local communities with these nodes as the core. Second, although the nodes at the top of the ranking have a large influence scope, they can easily attract adjacent nodes to their community because of their strong node influence, so it is easy to obtain local communities with these nodes as the core. Third, for the nodes at the middle of the ranking, there may be multiple adjacent nodes with the same node influence. Therefore, these nodes with the same node influence will be bypassed in the community expansion process, which leads to the irregularity of the resulting community, and it fails to form a local community with these nodes at the middle of the ranking as the core. From the Table 7 shows that *SOLCD* has the highest mean on all *DBLP* networks and the lowest *CV* except on *DBLP(1)*, which illustrates that compared with the other algorithms, *SOLCD* is more stable in terms of the seed dependence problem. However, the *SD* of *SOLCD* on some networks is high, which indicates that *SOLCD* still has a certain seed dependence problem. The other six comparison algorithms show sharp fluctuations in all subfigures of Fig. 12 and have small *SD* only on individual network in Table 7 which proves that all these algorithms have serious seed dependence problems.

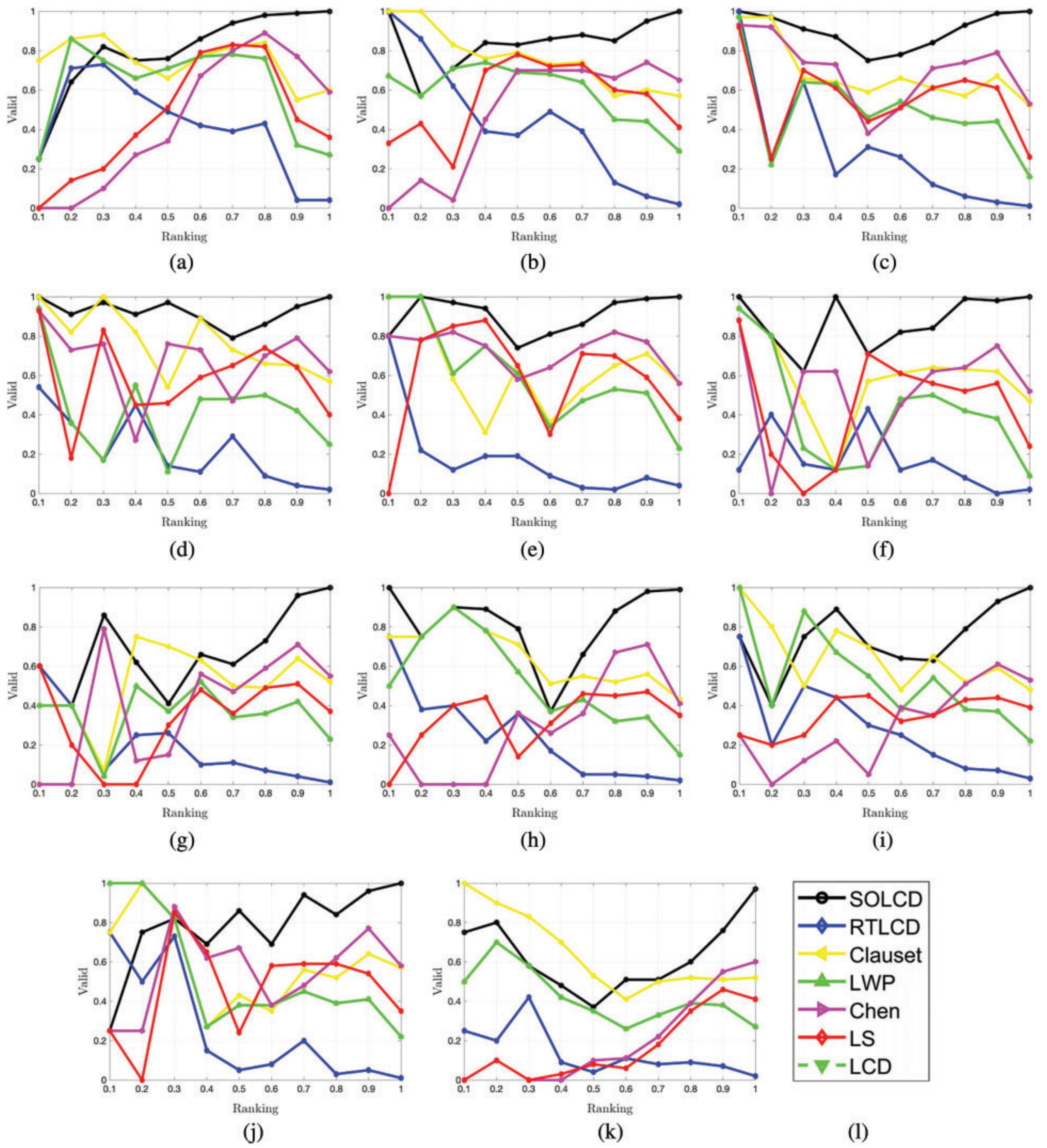
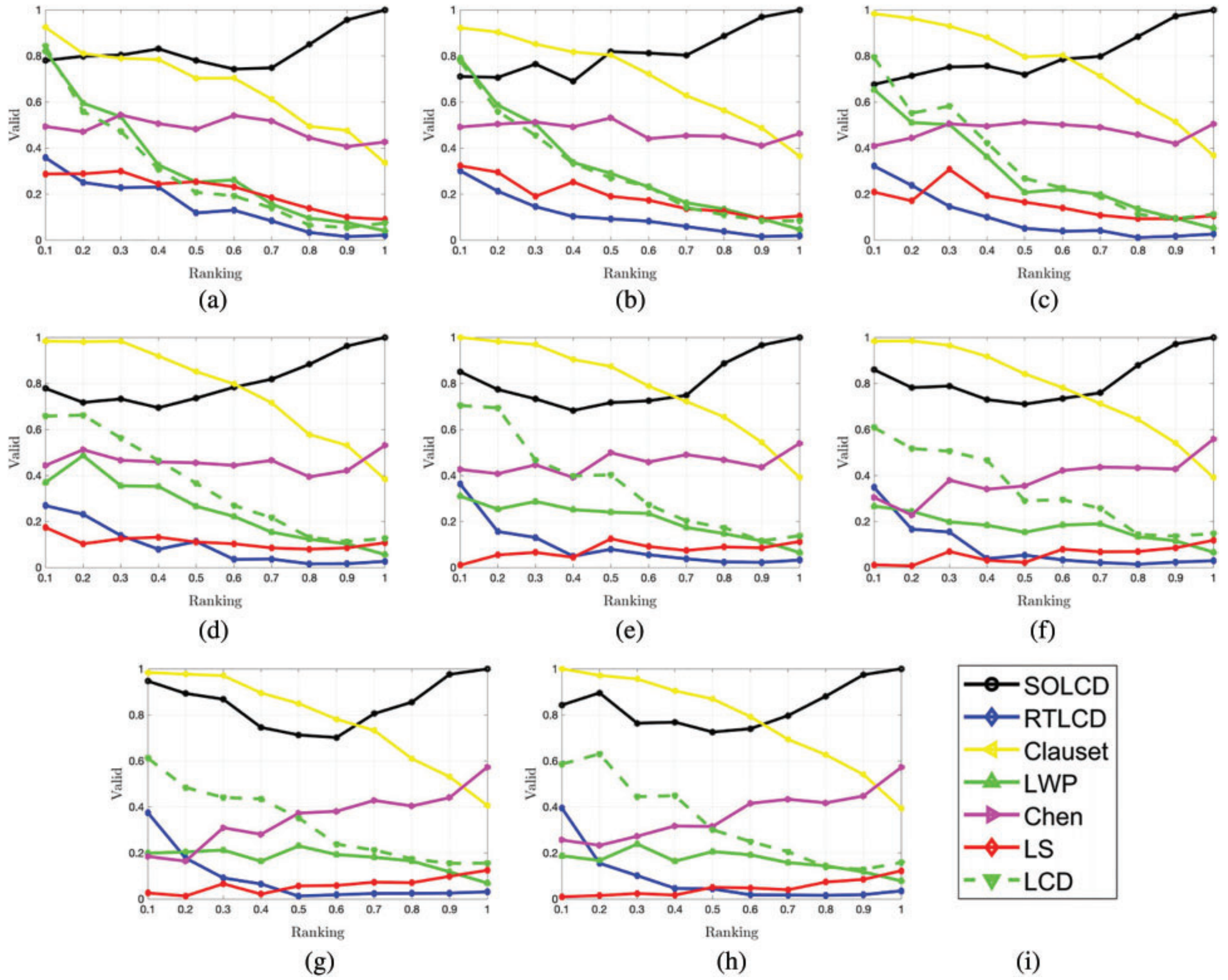


Figure 12: (a–l) The distribution of the valid seed-oriented local communities detected by the algorithms on a group of *DBLP* networks

Table 7: The standard deviation (SD), arithmetic mean and coefficient of variation (CV) of the proportion of the valid seed-oriented local communities detected by all the comparison algorithms on a group of DBLP networks

Network (ID)	Criteria	SOLCD	RTLCD	Clauset	LWP	Chen	LS
DBLP(1)	SD	0.2271	0.2430	**0.1121**	0.2359	0.3429	0.2936
	Mean	**0.7990**	0.4090	0.7470	0.6130	0.4430	0.4470
	CV	0.2843	0.5942	**0.1500**	0.3848	0.7740	0.6569
DBLP(2)	SD	**0.1310**	0.3251	0.1564	0.1477	0.3008	0.1938
	Mean	**0.8490**	0.4330	0.7590	0.5880	0.4780	0.5490
	CV	**0.1543**	0.7508	0.2060	0.2512	0.6294	0.3530
DBLP(3)	SD	**0.0914**	0.3112	0.1567	0.2272	0.1767	0.2021
	Mean	**0.9040**	0.2850	0.6860	0.4950	0.6980	0.5560
	CV	**0.1011**	1.0918	0.2284	0.4589	0.2531	0.3635
DBLP(4)	SD	**0.0667**	0.1795	0.1654	0.2337	0.1851	0.2217
	Mean	**0.9250**	0.2210	0.7680	0.4260	0.6760	0.5870
	CV	**0.0721**	0.8124	0.2153	0.5486	0.2739	0.3777
DBLP(5)	SD	**0.0967**	0.2299	0.2294	0.2532	0.0974	0.2777
	Mean	**0.9080**	0.1780	0.6350	0.6050	0.7270	0.5840
	CV	**0.1065**	1.2914	0.3612	0.4185	0.1340	0.4754
DBLP(6)	SD	**0.1387**	0.1442	0.2172	0.2865	0.2680	0.2837
	Mean	**0.8760**	0.1610	0.5860	0.4100	0.5240	0.4400
	CV	**0.1584**	0.8955	0.3706	0.6988	0.5115	0.6447
DBLP(7)	SD	0.2060	0.1882	0.1953	**0.1381**	0.2977	0.2086
	Mean	**0.6850**	0.1910	0.5100	0.3580	0.3940	0.3310
	CV	**0.3008**	0.9855	0.3830	0.3859	0.7557	0.6301
DBLP(8)	SD	0.1931	0.2322	**0.1512**	0.2373	0.2578	0.1564
	Mean	**0.8210**	0.2440	0.6460	0.5110	0.3020	0.3270
	CV	**0.2352**	0.9518	**0.2341**	0.4644	0.8538	0.4781
DBLP(9)	SD	0.1724	0.2275	0.1717	0.2464	0.2101	**0.0928**
	Mean	**0.7480**	0.2770	0.6500	0.5390	0.3030	0.3520
	CV	**0.2305**	0.8215	0.2641	0.4571	0.6935	0.2637
DBLP(10)	SD	0.2150	0.2925	0.2213	0.2934	**0.2098**	0.2492
	Mean	**0.7800**	0.2550	0.5910	0.5320	0.5500	0.4640
	CV	**0.2756**	1.1471	0.3745	0.5516	0.3815	0.5370
DBLP(11)	SD	0.1820	**0.1215**	0.2020	0.1390	0.2277	0.1753
	Mean	**0.6330**	0.1370	0.6420	0.4180	0.2070	0.1670
	CV	**0.2875**	0.8867	0.3146	0.3326	1.1000	1.0498

Fig. 13 shows that the curves of each algorithm in different figures are basically consistent. That is, the experimental results are not affected by parameter μ. The performance of *SOLCD* improves as the node ranking decreases, and the overall performance is excellent. *Clauset* has perfect performance on the top node ranking, but its performance drops rapidly as the node ranking decreases. The performance of *LCD* also drops rapidly as the node ranking decreases, but the overall performance is not as good as that of *Clauset*. The curves of *RTLCD*, *LWP* and *LS* are always at the bottom. The performance of *Chen* is very stable and does not fluctuate with the node ranking. Table 8 shows that *SOLCD* achieves the highest mean and a slightly higher *SD* and *CV* than Chen and LS. *Chen* achieves the lowest *SD* and *CV*, but its mean is much lower than that of *SOLCD*. *LS* performs excellently on SD; however, its *CV* is high, and the mean is low. The other algorithms have poor performance on the mean, *SD* or *CV*. The experimental results show that *SOLCD* basically has no seed dependence problem on *LFR*-μ artificial networks. Chen has no seed dependence problem, but its seed-oriented community detection ability is much weaker than that of *SOLCD*. *LS* has no seed dependence problem, but has no ability to detect the seed-oriented community. The other algorithms are seriously affected by seed dependence problems.



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

Figure 13: (a–i) The distribution of the valid seed-oriented local communities detected by the algorithms on a group of *LFR*-μ artificial networks

Table 8: The standard deviation (SD), arithmetic mean and coefficient of variation (CV) of the proportion of the valid seed-oriented local communities detected by all the comparison algorithms on a group of *LFR*-μ artificial networks

Network	Criteria	SOLCD	RTLCD	Clauset	LWP	Chen	LS	LCD
LFR-μ=0.1	SD	0.0857	0.1157	0.1818	0.2576	**0.0467**	0.0792	0.2591
	Mean	**0.8287**	0.1463	0.6632	0.3155	0.4828	0.2111	0.2910
	CV	0.1034	0.7910	0.2741	0.8163	**0.0968**	0.3751	0.8906
LFR-μ=0.2	SD	0.1074	0.0910	0.1885	0.2405	**0.0372**	0.0795	0.2313
	Mean	**0.8157**	0.1061	0.7058	0.3170	0.4745	0.1877	0.3040
	CV	0.1316	0.8579	0.2671	0.7586	**0.0783**	0.4233	0.7608
LFR-μ=0.3	SD	0.1107	0.1055	0.2054	0.2027	**0.0383**	0.0670	0.2408
	Mean	**0.8055**	0.0986	0.7548	0.2932	0.4735	0.1578	0.3347
	CV	0.1374	1.0700	0.2721	0.6912	**0.0809**	0.4247	0.7193
LFR-μ=0.4	SD	0.1052	0.0915	0.2138	0.1403	0.0397	**0.0280**	0.2183
	Mean	**0.8103**	0.0960	0.7722	0.2484	0.4588	0.1103	0.3566
	CV	0.1299	0.9530	0.2768	0.5646	**0.0866**	0.2542	0.6122
LFR-μ=0.5	SD	0.1113	0.1046	0.2041	0.0791	0.0448	**0.0335**	0.2162
	Mean	**0.8079**	0.0948	0.7826	0.2077	0.4559	0.0749	0.3564
	CV	0.1378	1.1034	0.2608	0.3807	**0.0982**	0.4471	0.6065
LFR-μ=0.6	SD	0.1022	0.1066	0.2031	0.0593	0.0894	**0.0364**	0.1747
	Mean	0.8211	0.0880	0.7759	0.1734	0.3881	0.0560	0.3363
	CV	**0.1245**	1.2108	0.2618	0.3419	0.2304	0.6503	0.5196
LFR-μ=0.7	SD	0.0968	0.1181	0.2048	0.0454	0.1065	**0.0362**	0.1863
	Mean	**0.8384**	0.0843	0.7744	0.1650	0.3676	0.0481	0.3287
	CV	**0.1155**	1.4002	0.2645	0.2753	0.2899	0.7519	0.5666
LFR-μ=0.8	SD	0.1073	0.1138	0.2020	0.0482	0.1232	**0.0348**	0.1614
	Mean	**0.8500**	0.0841	0.7733	0.1733	0.3535	0.0605	0.3255
	CV	**0.1262**	1.3525	0.2612	0.2783	0.3484	0.5752	0.4959

Fig. 14 and Table 9 show that the curves and data are roughly the same as those in Fig. 13 and Table 8. Therefore, we can conclude that the performance of algorithms is similar on *LFR*-α_{size} artificial networks and *LFR*-μ artificial networks.

Fig. 15 shows that *SOLCD* still performs stably, as in the above two groups of experiments. That is, the performance of *SOLCD* does not change as the parameter α_{degree} increases. However, the curves of the other algorithms decrease sharply as the parameter α_{degree} increases. The reason for this phenomenon is as follows. Table 10 shows that *SOLCD* achieves the highest mean and lowest *CV* on most networks and a slightly higher *SD* than *RTLCD*, Chen and LS. However, the mean of *RTLCD* and LS is so low that the results have no reference value. Chen performs well with a low parameter α_{degree} but worsens the parameter α_{degree} increases. The experimental results show that *SOLCD* basically has no seed dependence problem on *LFR*-α_{degree} artificial networks.

The other algorithms are seriously affected by seed dependence problems and are seriously affected by parameter α_{degree}.

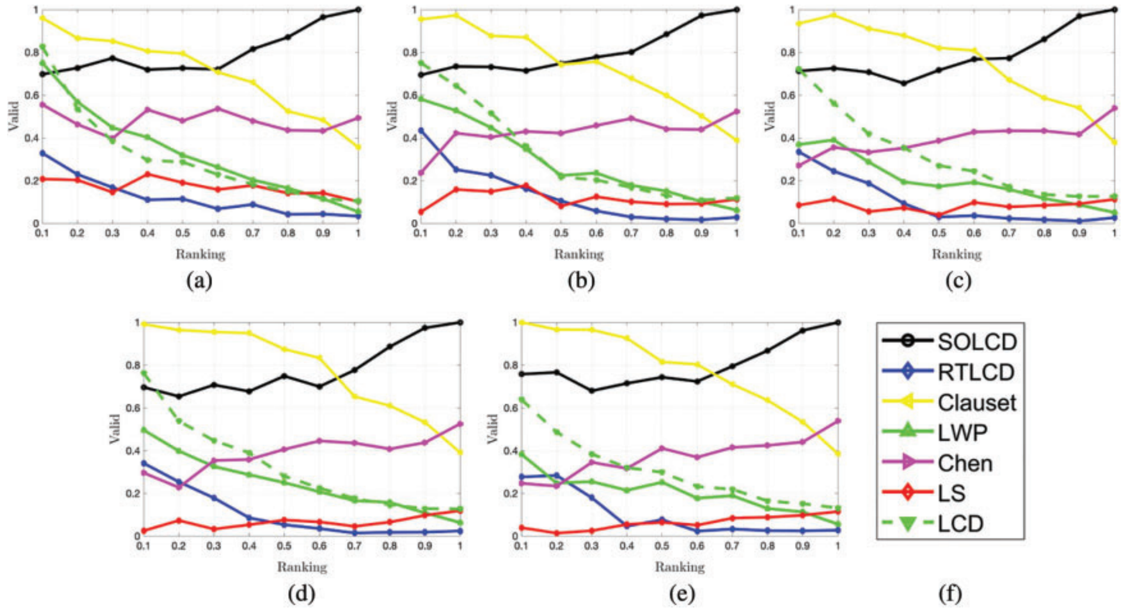
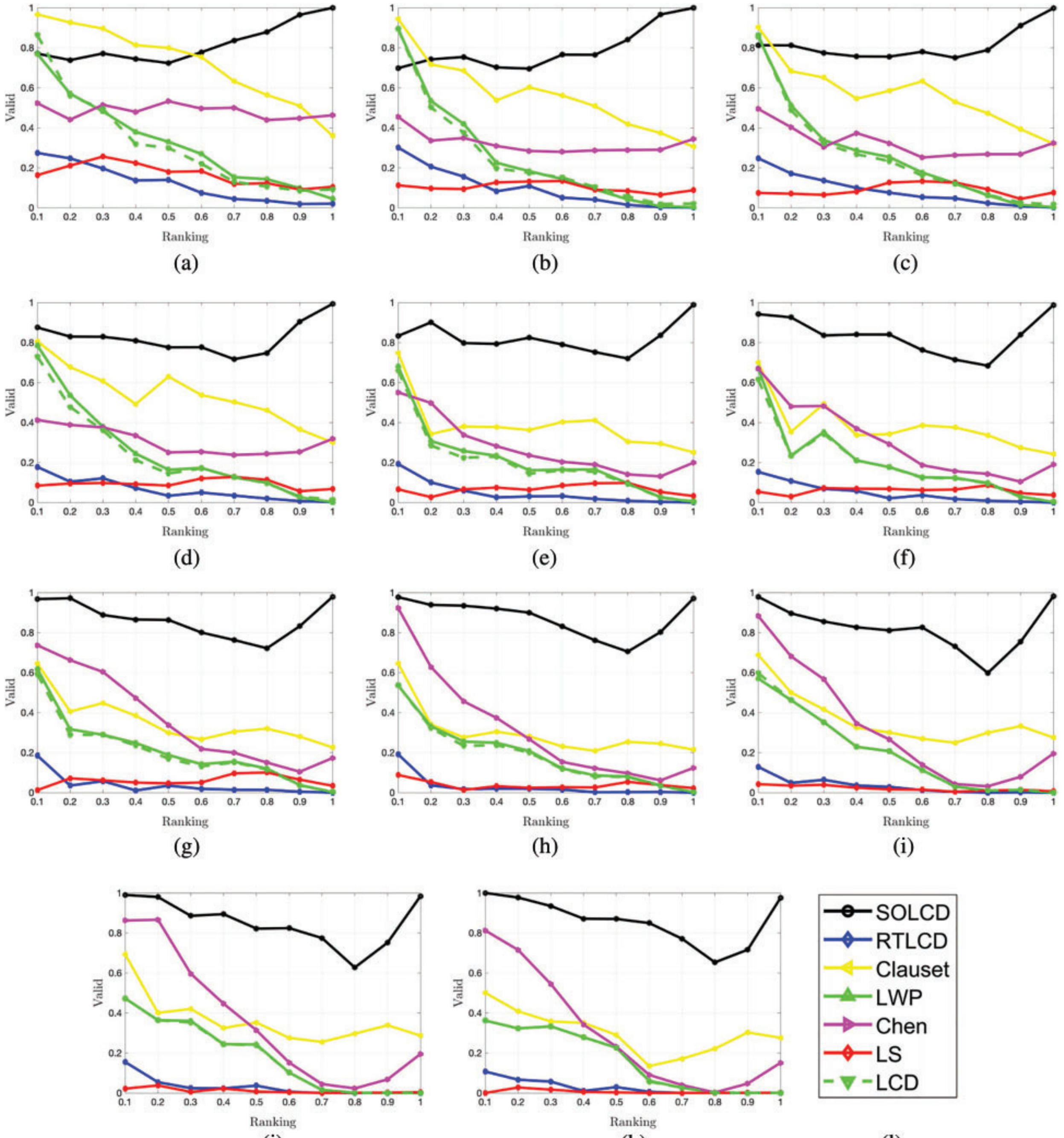


Figure 14: (a–f) The distribution of the valid seed-oriented local communities detected by the algorithms on a group of *LFR*-α_{size} artificial networks

Table 9: The standard deviation (SD), arithmetic mean and coefficient of variation (CV) of the proportion of the valid seed-oriented local communities detected by all the comparison algorithms on a group of *LFR*-α_{size} artificial networks

Network	Criteria	SOLCD	RTLCD	Clauset	LWP	Chen	LS	LCD
LFR-$\alpha_{\text{size}} = (5, 50)$	SD	0.1090	0.0947	0.1933	0.2166	0.0506	**0.0389**	0.2243
	Mean	**0.8009**	0.1223	0.7007	0.3286	0.4803	0.1694	0.3108
	CV	0.1361	0.7746	0.2759	0.6593	**0.1053**	0.2295	0.7219
LFR-$\alpha_{\text{size}} = (10, 100)$	SD	0.1091	0.1369	0.1941	0.1814	0.0757	**0.0385**	0.2355
	Mean	**0.8054**	0.1327	0.7340	0.2853	0.4265	0.1136	0.3216
	CV	**0.1354**	1.0320	0.2644	0.6357	0.1776	0.3387	0.7324
LFR-$\alpha_{\text{size}} = (15, 150)$	SD	0.1168	0.1149	0.1971	0.1139	0.0728	**0.0233**	0.2029
	Mean	**0.7884**	0.0999	0.7500	0.2013	0.3943	0.0825	0.3126
	CV	**0.1481**	1.1507	0.2628	0.5660	0.1846	0.2827	0.6491
LFR-$\alpha_{\text{size}} = (20, 200)$	SD	0.1262	0.1159	0.2126	0.1340	0.0842	**0.0283**	0.2108
	Mean	**0.7818**	0.1021	0.7753	0.2459	0.3892	0.0652	0.3221
	CV	**0.1615**	1.1358	0.2741	0.5450	0.2164	0.4344	0.6546
LFR-$\alpha_{\text{size}} = (25, 250)$	SD	0.1077	0.1062	0.2059	0.0917	0.0924	**0.0326**	0.1622
	Mean	**0.8007**	0.0996	0.7741	0.2017	0.3738	0.0632	0.3026
	CV	**0.1345**	1.0668	0.2660	0.4549	0.2472	0.5155	0.5362



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 15: (a–l) The distribution of the valid seed-oriented local communities detected by the algorithms on a group of *LFR*-α_{degree} artificial networks

Table 10: The standard deviation (SD), arithmetic mean and coefficient of variation (CV) of the proportion of the valid seed-oriented local communities detected by all the comparison algorithms on a group of *LFR*-α_{degree} artificial networks

Network	Criteria	SOLCD	RTLCD	Clauset	LWP	Chen	LS	LCD
LFR-$\alpha_{\text{degree}} = (5, 25)$	SD	0.0973	0.0952	0.1990	0.2312	**0.0348**	0.0551	0.2545
	Mean	**0.8202**	0.1183	0.7218	0.3235	0.4832	0.1652	0.3165
	CV	0.1187	0.8049	0.2757	0.7145	**0.0721**	0.3338	0.8040
LFR-$\alpha_{\text{degree}} = (6, 30)$	SD	0.1092	0.0985	0.1863	0.2847	0.0534	**0.0229**	0.2746
	Mean	**0.7924**	0.0961	0.5651	0.2540	0.3219	0.1018	0.2491
	CV	**0.1378**	1.0255	0.3296	1.1209	0.1659	0.2250	1.1021
LFR-$\alpha_{\text{degree}} = (7, 35)$	SD	0.0797	0.0789	0.1631	0.2619	0.0769	**0.0299**	0.2587
	Mean	**0.8136**	0.0860	0.5711	0.2620	0.3268	0.0884	0.2559
	CV	**0.0980**	0.9168	0.2856	0.9994	0.2351	0.3380	1.0108
LFR-$\alpha_{\text{degree}} = (8, 40)$	SD	0.0816	0.0566	0.1491	0.2468	0.0675	**0.0225**	0.2244
	Mean	**0.8255**	0.0621	0.5378	0.2531	0.3065	0.0938	0.2362
	CV	**0.0989**	0.9109	0.2772	0.9750	0.2201	0.2395	0.9503
LFR-$\alpha_{\text{degree}} = (9, 45)$	SD	0.0759	0.0592	0.1362	0.1912	0.1446	**0.0240**	0.1848
	Mean	**0.8233**	0.0470	0.3871	0.2093	0.2764	0.0660	0.1978
	CV	**0.0922**	1.2604	0.3519	0.9135	0.5232	0.3643	0.9342
LFR-$\alpha_{\text{degree}} = (10, 50)$	SD	0.0974	0.0501	0.1294	0.1933	0.1871	**0.0172**	0.1779
	Mean	**0.8368**	0.0473	0.3837	0.2023	0.3071	0.0588	0.1950
	CV	**0.1164**	1.0583	0.3371	0.9558	0.6092	0.2929	0.9125
LFR-$\alpha_{\text{degree}} = (11, 55)$	SD	0.0893	0.0547	0.1216	0.1747	0.2347	**0.0266**	0.1668
	Mean	**0.8651**	0.0373	0.3577	0.2115	0.3656	0.0586	0.2011
	CV	**0.1032**	1.4652	0.3400	0.8257	0.6419	0.4540	0.8293
LFR-$\alpha_{\text{degree}} = (12, 60)$	SD	0.0935	0.0578	0.1280	0.1613	0.2800	**0.0219**	0.1604
	Mean	**0.8739**	0.0305	0.3000	0.1903	0.3202	0.0376	0.1842
	CV	**0.1070**	1.8917	0.4265	0.8479	0.8743	0.5833	0.8707
LFR-$\alpha_{\text{degree}} = (13, 65)$	SD	0.1155	0.0406	0.1364	0.2049	0.2941	**0.0137**	0.2105
	Mean	**0.8258**	0.0322	0.3654	0.1985	0.3229	0.0204	0.2013
	CV	**0.1399**	1.2601	0.3733	1.0324	0.9108	0.6720	1.0459
LFR-$\alpha_{\text{degree}} = (14, 70)$	SD	0.1173	0.0477	0.1272	0.1795	0.3237	**0.0125**	0.1784
	Mean	**0.8533**	0.0298	0.3644	0.1801	0.3567	0.0103	0.1791
	CV	**0.1375**	1.5979	0.3491	0.9969	0.9076	1.2125	0.9962
LFR-$\alpha_{\text{degree}} = (15, 75)$	SD	0.1173	0.0373	0.1099	0.1561	0.2959	**0.0093**	0.1565
	Mean	**0.8617**	0.0275	0.3011	0.1612	0.2975	0.0055	0.1607
	CV	**0.1361**	1.3549	0.3649	0.9688	0.9945	1.6973	0.9737

4.8 The Discussion of the Experimental Results

Based on the above experimental results, we can obtain the following conclusions. First, the proposed algorithm has a great ability to detect high-quality seed-oriented local communities among real-world networks, which proves that *SOLCD* solves the seed-deviation problem. Second, the seed-oriented local community detection ability of the proposed algorithm is not affected by parameters μ, α_{size} and α_{degree}. Third, *SD* and *CV* are the proportions of valid seed-oriented local communities detected by *SOLCD* on real-world networks and artificial networks, respectively. *SOLCD* achieves a low *SD* and *CV*, which proves that *SOLCD* can detect high-quality seed-oriented local communities with nodes with different node influences as seeds. This illustrates that *SOLCD* solves seed dependence problems. In addition, *SOLCD* achieves excellent results on groups of artificial networks with different parameters, which shows that *SOLCD* has strong robustness. However, there are still problems to be solved. *SOLCD* still does not completely resolve the seed dependence problem, especially when taking the nodes with medium node influence as the seed.

5 Conclusion

Research on local community detection has achieved excellent achievements. However, there are still some problems to be solved, such as the seed deviation problem, the seed dependence problem and the parameter dependence problem. In order to solve these problems, this paper proposes a seed-oriented local community detection algorithm, named *SOLCD*, based on influence spreading. To solve the seed deviation problem and the seed dependence problem, we propose a seed-oriented algorithm, which always takes the given node as the seed, and always takes the seed as the influence core in the community expansion process. To solve the parameter dependence problem, we propose a local community detection algorithm based on influence spreading without any parameters. In addition, we propose a local community effectiveness index (*LCE*) and a local community uniqueness index (*LCE*) to estimate the quality of seed-oriented local communities. Efficient and rapid detection of seed-oriented communities can improve the accuracy of personalized recommendation of goods and information and help public opinion analysis.

This paper compares *SOLCD* with six other state-of-the-art local community detection algorithms on *LFR* artificial networks and real-world networks. The experimental results show that *SOLCD* has a great ability to detect high-quality seed-oriented local communities among real-world networks, which proves that *SOLCD* solves the seed deviation problem. Taking nodes with different node influences as seeds, *SOLCD* can detect high-quality seed-oriented local communities, which illustrates that *SOLCD* solves the seed dependence problem. In addition, *SOLCD* achieves excellent results on groups of artificial networks with different parameters, which shows that *SOLCD* has strong robustness.

However, there are still problems to be solved. *SOLCD* still has not completely resolved the seed dependence problem, especially when taking the nodes with medium node influence as the seed. We will focus on solving the seed dependence problem completely in future research.

Funding Statement: National Natural Science Foundation of China (Nos. 61672179, 61370083, 61402126), Heilongjiang Province Natural Science Foundation of China (No. F2015030), Science

Fund for Youths in Heilongjiang Province (No. QC2016083) and Postdoctoral Fellowship in Heilongjiang Province (No. LBH-Z14071).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Fang, Y., Huang, X., Qin, L., Zhang, Y., Zhang, W. et al. (2020). A survey of community search over big graphs. *The VLDB Journal, 29(1),* 353–392. DOI 10.1007/s00778-019-00556-x.
2. Mittal, R., Bhatia, M. (2021). Classification and comparative evaluation of community detection algorithms. *Archives of Computational Methods in Engineering, 28,* 1417–1428. DOI 10.1007/s11831-020-09421-5.
3. Pizzuti, C. (2018). Evolutionary computation for community detection in networks: A review. *IEEE Transactions on Evolutionary Computation, 22(3),* 464–483. DOI 10.1109/TEVC.2017.2737600.
4. Garza, S. E., Schaeffer, S. E. (2019). Community detection with the label propagation algorithm: A survey. *Physica A: Statistical Mechanics and its Applications, 534,* 122058. DOI 10.1016/j.physa.2019.122058.
5. Kanawati, R. (2015). Empirical evaluation of applying ensemble methods to ego-centred community identification in complex networks. *Neurocomputing, 150,* 417–427. DOI 10.1016/j.neucom.2014.09.042.
6. Zhang, R., Li, L., Bao, C. M., Zhou, L. H., Kong, B. (2015). The community detection algorithm based on the node clustering coefficient and the edge clustering coefficient. *11th World Congress on Intelligent Control and Automation*, pp. 3240–3245. Shengyang, China.
7. Peng, W., Jing, L. (2016). A multi-agent genetic algorithm for local community detection by extending the tightest nodes. *2016 IEEE Congress on Evolutionary Computation*, pp. 3215–3221. Vancouver, BC, Canada.
8. Liakos, P., Ntoulas, A., Delis, A. (2016). Scalable link community detection: A local dispersion-aware approach. *IEEE International Conference on Big Data (Big Data)*, pp. 716–725. Washington, DC, USA.
9. Zhu, J. R., Chen, B. L., Zeng, Y. F. (2020). Community detection based on modularity and k-plexes. *Information Sciences, 513,* 127–142. DOI 10.1016/j.ins.2019.10.076.
10. Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., Shir, E. (2007). From the cover: A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Science, 104(27),* 11150–11154. DOI 10.1073/pnas.0701175104.
11. Wang, Z. X., Li, Z. C., Ding, X. F., Tang, J. H. (2016). Overlapping community detection based on node location analysis. *Knowledge-Based Systems, 105,* 225–235. DOI 10.1016/j.knosys.2016.05.024.
12. Lancichinetti, A., Fortunato, S., Kertész, J. (2009). Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics, 11(3).* DOI 10.1088/1367-2630/11/3/033015.
13. Baumes, J., Goldberg, M. K., Krishnamoorthy, M. S., Magdon-Ismail, M., Preston, N. (2005). Finding communities by clustering a graph into overlapping subgraphs. *Proceedings of the IADIS International Conference on Applied Computing*, Algarve, Portugal.
14. Lee, C., Reid, F., Mcdaid, A., Hurley, N. (2010). Detecting highly overlapping community structure by greedy clique expansion. *Proceedings of the 4th SNAKDD Workshop*, pp. 33–42. USA.
15. Whang, J. J., Gleich, D. F., Dhillon, I. S. (2015). Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Transactions on Knowledge & Data Engineering, 28(5),* 1272–1284. DOI 10.1109/TKDE.2016.2518687.
16. Ding, X. Y., Zhang, J. P., Yang, J. (2018). A robust two-stage algorithm for local community detection. *Knowledge-Based Systems, 152,* 188–199. DOI 10.1016/j.knosys.2018.04.018.
17. Cheng, J. J., Zhang, W. B., Yang, H. J., Su, X., Ma, T. et al. (2020). A seed-expanding method based on topsis for community detection in complex networks. *Complexity, 2020,* 1–14. DOI 10.1155/2020/9017239.
18. Guo, K., Huang, X., Wu, L., Chen, Y. (2021). Local community detection algorithm based on local modularity density. *Applied Intelligence, (4),* 1–16. DOI 10.1007/s10489-020-02052-0.
19. Ni, L., Luo, W. J., Zhu, W. J., Hua, B. (2020). Local overlapping community detection. *ACM Transactions on Knowledge Discovery from Data, 14(1),* 1–25. DOI 10.1145/3361739.

20. Luo, W. J., Yan, Z. L., Bu, C. Y., Zhang, D. F. (2017). Community detection by fuzzy relations. *IEEE Transactions on Emerging Topics in Computing, 8(2),* 478–492. DOI 10.1109/TETC.6245516.
21. Kamuhanda, D., Wang, M., He, K. (2020). Sparse nonnegative matrix factorization for multiple local community detection. *IEEE Transactions on Computational Social Systems, 7(5),* 1220–1233. DOI 10.1109/TCSS.6570650.
22. Kloster, K., Gleich, D. F. (2014). Heat kernel based community detection. *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1386–1395. New York, USA.
23. Hu, Y. M., Yang, B., Wong, H. S. (2016). A weighted local view method based on observation over ground truth for community detection. *Information Sciences, 355,* 37–57. DOI 10.1016/j.ins.2016.03.028.
24. He, K., Sun, Y., Bindel, D., Hopcroft, J. E., Li, Y. (2015). Detecting overlapping communities from local spectral subspaces. *IEEE International Conference on Data Mining (ICDM)*, pp. 769–774. Atlantic City, NJ.
25. Yao, Y. T., Wu, W., Lei, M. T., Zhang, X. (2016). Community detection based on variable vertex influence. *IEEE First International Conference on Data Science in Cyberspace*, pp. 418–423. China.
26. You, X. M., Ma, Y. H., Liu, Z. Y. (2020). A three-stage algorithm on community detection in social networks. *Knowledge-Based Systems, 187,* 104822.1–104822.12. DOI 10.1016/j.knosys.2019.06.030.
27. Fortunato, S. (2009). Community detection in graphs. *Physics Reports, 486(3–5)*.
28. Yang, J., Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. *Knowledge & Information Systems, 42(1),* 181–213. DOI 10.1007/s10115-013-0693-z.
29. Raghavan, U. N., Albert, R., Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E Statistical Nonlinear and Soft Matter Physics, 76(3),* 036106. DOI 10.1103/PhysRevE.76.036106.
30. Xu, G. Q., Guo, J. W., Yang, P. L. (2020). TNS-IPA: An improved label propagation algorithm for community detection based on two-level neighbourhood similarity. *IEEE Access, 9,* 23526–23536. DOI 10.1109/access.2020.3045085.
31. Wu, Z. L., Wang, X., Fang, W. Y., Liu, L. Z., Tang, S. T. et al. (2020). Community detection based on first passage probabilities. *Physics Letters A, 390*. DOI 10.1016/j.physleta.2020.127099.
32. Gregory, S. (2009). Finding overlapping communities in networks by label propagation. *New Journal of Physics, 12(10),* 2011–2024. DOI 10.1088/1367-2630/12/10/103018.
33. Tang, M. L., Liu, Q., Ma, T. H., Cao, J., Tian, Y. et al. (2019). K-lowest-influence overlapping nodes based community detection in complex networks. *IEEE Access, 7,* 109646–109661. DOI 10.1109/access.2019.2930474.
34. Yi, Y. Q., Jin, L. H., Yu, H., Luo, H. R., Cheng, F. (2021). Density sensitive random walk for local community detection. *IEEE Access, 9,* 27773–27782. DOI 10.1109/access.2021.3058908.
35. Liu, J. X., Shao, Y. X., Su, S. (2021). Multiple local community detection via high-quality seed identification over both static and dynamic networks. *Data Science and Engineering, 6(3),* 249–264. DOI 10.1007/s41019-021-00160-6.
36. Aaron, C. (2005). Finding local community structure in networks. *Physical Review E Statal Nonlinear & Soft Matter Physics, 72(2),* 026132. DOI 10.1103/PhysRevE.72.026132.
37. Luo, F., Wang, J. Z., Promislow, E. (2008). Exploring local community structures in large networks. *Web Intelligence and Agent Systems, 6(4),* 387–400. DOI 10.3233/WIA-2008-0147.
38. Chen, J., Zaï, O. R., Goebel, R. (2009). Local community identification in social networks. *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, pp. 237–242. Athens, Greece.
39. Wu, Y. J., Han, H., Hao, Z. F., Chen, F. (2012). Local community detection using link similarity. *Journal of Computer Science & Technology, 27(6),* 1261–1268. DOI 10.1007/s11390-012-1302-4.
40. Meng, F. R., Zhu, M., Zhou, Y., Zhou, R. R. (2014). Local community detection in complex networks based on maximum cliques extension. *Mathematical Problems in Engineering, 2014,* 1–12. DOI 10.1155/2014/653670.

41. Batagelj, V., Zaversnik, M. (2003). An o(m) algorithm for cores decomposition of networks. *Computer Science, 1(6),* 34–37. DOI 10.1007/BF01074693.
42. Zhang, Z. (2013). Ranking spreaders by decomposing complex networks. *Physics Letters A, 377(14),* 1031–1035. DOI 10.1016/j.physleta.2013.02.039.
43. Liu, J. G., Ren, Z. M., Guo, Q. (2013). Ranking the spreading influence in complex networks. *Physica A Statistical Mechanics & its Applications, 392(18),* 4154–4159. DOI 10.1016/j.physa.2013.04.037.
44. Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L. et al. (2010). Identification of influential spreaders in complex networks. *Nature Physics, 6(11),* 888–893. DOI 10.1038/nphys1746.
45. Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research, 33(4),* 452–473. DOI 10.1086/jar.33.4.3629752.
46. Danon, L., Duch, J., Diaz-Guilera, A., Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics, 2005(9),* 09008. DOI 10.1088/1742-5468/2005/09/P09008.
47. Li, J., Wang, X., Wu, P. (2015). Review on community detection methods based on local optimization. *Bulletin of Chinese Academy of Science, 30(2),* 238–247.
48. Lancichinetti, A., Fortunato, S., Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E, 78(4),* 046110. DOI 10.1103/PhysRevE.78.046110.
49. Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E. et al. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology & Sociobiology, 54(4),* 396–405. DOI 10.1007/s00265-003-0651-y.
50. Krebs, V. (2004). Social network of political books. www.visualcomplexity.com.
51. Girvan, M., Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America, 99(12),* 7821–7826. DOI 10.1073/pnas.122653799.