Neighborhood coreness algorithm for identifying a set of influential spreaders in complex networks

Xiong YANG[1,2,3], De-Cai HUANG[1] and Zi-Ke ZHANG[4]
[1] School of Computer Science and Technology, Zhejiang University of Technology,
Hangzhou 310023 - China
[e-mail: yangx@cit.edu.cn, hdc@zjut.edu.cn]
[2] School of Computer and Information Engineering, Changzhou Institute of Technology,
Changzhou 213002 - China
[3] Shanghai Key Laboratory of Integrated Administration Technologies for Information Security,
Shanghai Jiao Tong University, Shanghai 200240 - China
[4] Alibaba Research Center for Complexity Sciences, Hangzhou 311121 - China
[e-mail: zhangzike@gmail.com]
*Corresponding author: De-Cai HUANG

Received January 9, 2017; revised March 5, 2017; accepted March 11, 2017; published June 30, 2017

Abstract

In recent years, there has been an increasing number of studies focused on identifying a set of spreaders to maximize the influence of spreading in complex networks. Although the *k-core* decomposition can effectively identify the single most influential spreader, selecting a group of nodes that has the largest *k-core* value as the seeds cannot increase the performance of the influence maximization because the propagation sphere of this group of nodes is overlapped. To overcome this limitation, we propose a neighborhood coreness cover and discount heuristic algorithm named "*NCCDH*" to identify a set of influential and decentralized seeds. Using this method, a node in the high-order shell with the largest neighborhood coreness and an uncovered status will be selected as the seed in each turn. In addition, the neighbors within the same shell layer of this seed will be covered, and the neighborhood coreness of the neighbors outside the shell layer will be discounted in the subsequent round. The experimental results show that with increases in the spreading probability, the *NCCDH* outperforms other algorithms in terms of the affected scale and spreading speed under the Susceptible-Infected-Recovered (*SIR*) and Susceptible-Infected (*SI*) models. Furthermore, this approach has a superior running time.

Keywords: Influential spreaders, influence maximization, complex networks, k-core decomposition, epidemic spreading

This work is supported by the National Natural Science Foundation of China(No.61673151,61503110); Natural Science Foundation in Zhejiang Province of China (Grant No. LQ13F030015,LY14A050001,LQ16F030006); Applied Basic Research for Science & Technology Projects in Changzhou City of Jiangsu Province, China (Grant No. CJ20159013); Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, China (Grant No. AGK201601); Natural Science Foundation of Jiangsu Higher Education Institutions, China (Grant No.16KJB520003)

https://doi.org/10.3837/tiis.2017.06.010 **ISSN : 1976-7277**

1. Introduction

Identifying the most influential spreaders has received significant attention in the field of network science in recent years [1-2] and is of considerable significance for controlling the outbreak of epidemics [3], enhancing the effects of e-commercial advertisements [4], preventing the disastrous collapse of traffic networks or the Internet [5], optimizing information dissemination [6], identifying excellent players in sporting competitions [7], and predicting papers and authors with potentiality in co-authorship and citation networks [8]. Many classical centrality methods based on the topological structure of networks have been widely applied for identifying influential spreaders, such as the *degree* centrality, *betweenness* centrality [9], *closeness* centrality [10], *Katz* centrality [11], etc. Recently, Kitsak et al. [12] proposed a *k-core* (also called the *k-shell*) centrality using the *k-core* decomposition in networks to estimate the spreading influence of a node. A node located in the core of the network likely has more influence than a node located in the periphery. In addition, the *k-core* centrality has certain limitations; for example, the method will divide many nodes with different spreading abilities into the same shell layer and consider only the residual degree of the node while ignoring the out-leaving links from the group to the nodes outside this layer. Several hypotheses have been proposed to resolve these problems. In 2013, Zeng et al. [13] proposed a mixed degree decomposition centrality that considers both the residual degree and the exhausted degree. Liu et al. [14] distinguished the influences of the nodes in the same shell by measuring the shortest distances from the target node to all the nodes located in the highest-order shell. The nodes whose locations are closer to the network core play a more significant role in the spreading process. In 2014, Pei et al. [15] discovered that the influential nodes are always located in the core of the network through various social platforms, such as *Twitter*, *Facebook*, and *Livejournal*. In the same year, Bae et al. [16] proposed a neighborhood coreness (abbreviated as *nc*) centrality, where the *nc* value of each node is defined as the sum of the *k-shell* indices of its neighbors. Such a method is able to reasonably evaluate the nodes in the periphery of the network and accurately identify the influential nodes because balance is achieved between the degree and the coreness of a node. In 2015, Liu et al. [17-18] introduced a measure based on the link diversity of shells to effectively distinguish the true core from the core-like groups, which helps enhance the ranking of the influential nodes. Similarly, Fu et al. [19] combined the global diversity of the shells and local features to identify the most influential nodes in a more fine-grained capacity. Indeed, the influential nodes play an important role in the application of the networks. In WSN, the influential node is usually called the Cluster Head (*CH*), and it is responsible for transferring data to the sink nodes. In 2017, PGV Naranjo and M Shojafar et al. [20] proposed a modified Stable Election Protocol (*SEP*) named Prolong-SEP (*P-SEP*) to select the influential node and prolong the stable period of sensor networks by maintaining a balanced energy consumption. The *P-SEP* divides the nodes into advanced and normal nodes according to nodes' heterogeneities. The influential node in this mechanism is not fixed.

Several works have considered network control and optimization. Pooranian et al. [21] proposed a hybrid-scheduling algorithm to solve the NP-hard problem in the task scheduling for grid computing. In addition, Pooranian et al. [22] also proposed two heuristic algorithms to minimize the operating power and pollution emission to convert the problem to a single-objective function. The influence maximization is a classical problem in network control and optimization. Although many works can investigate single vital nodes, selecting a

group of top-ranking influential spreaders does not produce the most satisfactory spreading results because the propagation spheres of these seeds may overlap. Kempe et al. [23] proved that the optimization problem of finding seed nodes to maximize their influence [24] is a NP-complete problem and proposed a greedy algorithm to reach near optimal solutions. However, this greedy algorithm has a high time complexity and is unsuitable for large-scale networks. Leskovec et al. [25] proposed an improved greedy algorithm *CELF*, which selects seed nodes to reduce the running times of the influence maximization. However, the efficiency of the algorithm is still relatively low. Therefore, most works in recent years have turned to heuristic algorithms. Chen et al. [26] proposed a degree discount heuristic (*DDH*) algorithm, which assumes that if a neighbor of a node is selected as the seed, the degree of this node should be discounted in the calculation. In 2014, Sankar et al. [27] first used the diffusion degree heuristic (*DiDH*) algorithm in large-scale social networks to develop an influence maximization scheme based on the *IC* model. This method not only considers the 2-step neighborhood information but also analyzes the impact of the propagation probability. Kim et al. [28] developed an influence maximization scheme by choosing influential neighbors. This method used the local neighborhood information to evaluate the propagation ability at a more realistic level. In 2016, Liu et al. [29] found that the 2-step neighborhood information could significantly improve the performance of the influence evaluation. This conclusion is of great significance for heuristic influence maximization algorithms. Recently, Zhang et al. [30] noted that certain spreaders were so close together that they overlapped the spreading sphere; therefore, they proposed a simply yet effectively iterative method called *VoteRank* to identify a set of decentralized spreaders with the best spreading ability. In terms of influence maximization by *k-core* decomposition, Kitsak et al. [12] proposed the maximum core cover (*MCC*) method for selecting the decentralized seeds by covering the neighbors of a seed located in the high-order shell. In 2015, Cao et al. [31] proposed an improved core cover algorithm (*CCA*) that combines the shell information and the node degree that selects the node located in the high-order shell with the largest degree as the seed and then covers all neighbors of this seed. Unfortunately, the *MCC* and *CCA* are still inadequate as shown in **Fig. 1**. The *MCC* will randomly select one of the nodes from *A* to *D* in the highest shell layer as the seed, which is a highly coarse-grained approach. The *CCA* will randomly select either *A* or *B* with the largest degree in the highest shell as the first seed. *A* and *B* have more link diversity in the shells than *C* and *D*. Node *A* is likely chosen as the first seed by *CCA* and its neighbors from *B* to *E* are all covered, and then node *L* is selected as the second seed. However, because one path is available between nodes *A* and *E*, the nodes from *F* to *J* will never be activated once node *A* fails to infect node *E* with larger degrees. Based on this discussion, the fewer common neighbors between node *A* inside the highest shell and the neighbor *E* outside this shell indicates that simply covering all neighbors of the seed located in the high-order shell will not generate the most influential spreading results.

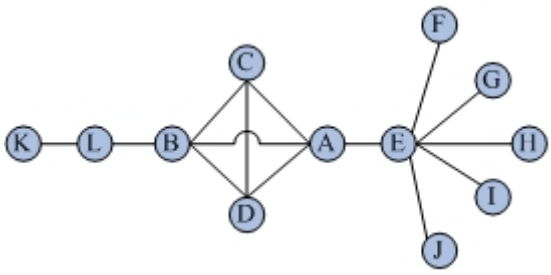


Fig. 1. Sample network for seed selection under the cover effect.

In this paper, we propose a simply yet effective *nc* cover and discount heuristic algorithm (*NCCDH*) to identify a set of decentralized spreaders with the best spreading ability. The *NCCDH* selects the seeds according to the *nc* centrality proposed in [16]. After choosing a node from the high-order shell as the seed in each turn, the neighbors within the same shell layer of this seed will be covered. At the same time, the *nc* of the neighbors outside this shell will be discounted in the subsequent round. The *NCCDH* can also provide a reasonable assessment for the nodes with a large degree located in the periphery of the network. The experimental results on real datasets show that the *NCCDH* outperforms traditional methods in terms of the affected scale and spreading speed. More importantly, the time complexity of this approach is superior to that of traditional methods.

2. Preliminaries

For a network $G(V,E)$, V and E are the set of nodes and set of edges, respectively, and the number of nodes is indicated by $n=|V|$, the number of edges is indicated by $m=|E|$, and the graph G is represented by the adjacency matrix $A=\{a_{ij}\}$. If an edge occurs between node i and node j, then $a_{ij}=1$; otherwise, the value is 0. The set of neighbors for node i is represented as $N(i)$.

2.1 *k*-core centrality

The *k-core* centrality is determined by the *k-core* decomposition, which will remove all the nodes with a degree =1 in the first step. Such activity will reduce the degrees of the remaining nodes so that all the nodes with degrees ≤1 are successively deleted until the degree of all the remaining nodes is >1. All the removed nodes are divided into the 1-shell, and the *k-core* centrality of these nodes is equal to 1. Secondly, all the remaining nodes with degrees =2 will be removed according to the above steps, and the nodes whose residual degrees ≤2 are also continually deleted until the degrees of all the remaining nodes >2. All the removed nodes in this step are divided into the 2-shell and the *k-core* centrality of these nodes is equal to 2. This decomposition process will continue until all the nodes are assigned into the corresponding shell layer. The *k-core* centrality of node i is denoted by $C_{KC}(i)$, which equals its corresponding shell indices, denoted by $kc(i)$, and is shown as following formula:

$$C_{KC}(i) = kc(i) \tag{1}$$

2.2 Neighborhood coreness centrality

Many works has been done to discriminate the spreading ability of nodes in the same *k-shell*. Zeng et al. [13] proposed the mixed degree decomposition method, which alters the *k-core* decomposition process by considering both the residual degree k_r and the exhausted degree k_e, as follows:

$$C_{km}(i) = k_r(i) + \lambda * k_e(i) \tag{2}$$

where $C_{km}(i)$ is defined as the mixed degree of node i, λ is a tunable parameter between 0 and 1. However, the parameter λ should be determined by the structure of a given network, and it is difficult to find the optimal parameter λ to achieve better result. Moreover, this method gives equal importance to the removed nodes regardless of whether they reside in the core or in the periphery of the network.

Inspired by these previous studies,the *nc* centrality value considers not only the degree of the node but also the shell layer of its neighbors into account. Therefore, a spreader with more connections to the neighbors located in the core of the network will be more influential. The *nc*

centrality of node i is denoted by $C_{nc}(i)$ and is shown as follows:

$$C_{nc}(i) = \sum_{j \in N(i)} C_{KC}(j) \tag{3}$$

2.3 Propagation models

The *SIR* model is widely used for information dissemination and disease diffusion and in various fields. In this paper, the model is applied to estimate the spreading influence of different methods. The *SIR* model consists of three states: *Susceptible*, *Infected*, and *Recovered*. The *Susceptible* set nodes are susceptible to information or diseases. The *Infected* set nodes are the nodes that are already infected or activated by diseases and information. The *Recovered* set nodes represent the nodes that have been immunized or recovered and will never be infected again. At each time step t, the *Infected* nodes attempt to infect their neighbors whose status are *Susceptible* with an infection probability of β. Then, each *Infected* node attempts to be recovered with an immune probability of γ. If an infected node is successfully recovered, the status of this node will be converted from *Infected* to *Recovered* and the node will never be infected again. The propagation of the *SIR* model will terminate when there are no infected nodes in the network. Similar to the *SIR*, the *SI* model contains only two states: *Susceptible* and *Infected*. The propagation in the *SI* model will terminate when there are no more susceptible nodes to be infected in the network. The spreading influence of the seed u at time t in the *SIR* and *SI* models is defined as $F_u(t)$:

$$F_u(t) = \begin{cases} I_u(t) + R_u(t) & in \quad SIR \\ I_u(t) & in \quad SI \end{cases} \tag{4}$$

where $I_u(t)$ and $R_u(t)$ are the number of infected and recovered nodes at time t, respectively, and they originated from the initial seed u.

2.4 Problem definition for influence maximization

$F_u(t)$ represents the spreading influence of the seed u at time t according to the propagation model. The input is the topology of the network G and the number of seeds k. The optimal objective is to maximize the spreading influence in the networks, which is defined as searching a set S consisting of k seeds to maximize $|\bigcup_{u \in S} F_u(t)|$. Finally, the output is a set of seed nodes.

- **Objective**: $Max\{F_S(t)=|\bigcup_{u \in S} F_u(t)|\}$
- **Input**: the network $G(V,E)$; the number of seeds k
- **Constraints:** $i \in [1,k]$; $S_i=S_{i-1} \cup \{u\}$; $u \in V \backslash S_{i-1}$; $|S|=k$
- **Output:** seeds set S

The greedy algorithm is proposed for this problem, and it starts from a seed set $S_0=\emptyset$ and selects node $u=argmax(|F_{S_{i-1} \cup u}|-|F_{S_{i-1}}|)$ as a seed node in step i. The greedy algorithm traverses all uninfected nodes when selecting a seed in each step, thereby resulting in a high time complexity. Therefore, many studies have resorted to using heuristic algorithms.

3. Heuristic algorithm for influence maximization by neighborhood coreness

3.1 *NCCDH* algorithm

Because the traditional *k-core* centrality method divides several nodes with different spreading abilities into the same shell layer, this paper promotes the ranking of influential spreaders based on the *nc* centrality. The *nc* centrality uses the *k-core* centrality of its neighbors to estimate the spreading influence of a node, which can improve the ranking of spreaders at a more fine-grained level. Based on this advantage, we propose a *nc* cover and discount heuristic algorithm (*NCCDH*) to maximize the spreading influence. The main concept of this algorithm is to select the seeds according to the *nc* indices of the nodes (observed in Section 2.2), which chooses the node with the largest *nc* and uncovered status as the target in the current round. If node *u* is selected as the seed, then the status of *u* will be marked as covered ($COVER_u=true$), and its neighbors within the same shell layer will also be covered. Similar ideas are also used in the *CCA* and *MCC* because the nodes in the high-order shell are closely clustered to each other, which results in overlapping spheres of influence spreading. However, the *NCCDH* uses the *nc* centrality to select seeds, which indicates that this method has a better accuracy than the *CCA* and *MCC*. More importantly, compared with the *CCA* and *MCC* that cover all the neighbors, the *NCCDH* does not cover the neighbors outside the shell but rather discounts the *nc* value of such nodes, which prevents the neighbors outside the shell that have larger degrees but fewer common neighbors with the seeds inside the shell from being directly covered. The detailed process is shown in **Algorithm 1**, where $C_{nc}(u)$ denotes the *nc* centrality of node *u*, $COVER_u$ denotes the covered status of node *u*, and $C_{KC}(u)$ denotes the *k-core* centrality of node *u*. As shown in **Fig. 1**, an assumption of the process is that node *A* will be selected as the first seed, which results in $C_{nc}(E)=8-3=5>C_{nc}(L)=4$. Therefore, the *NCCDH* will select *E* as the second seed to spread to the larger scale.

Algorithm 1. *NCCDH algorithm*

Input: $G(V,E)$, k
Output: *Seed sets S*

```
S=∅
for each vertex u ∈V do
   calculate C_kc(u) according to formula (1)
   calculate C_nc(u) according to formula (2)
   COVER_u=false
end for
while |S|<k do
    v=argmax_u{C_nc(u)| u ∈V\S, COVER_u=false}
    S=S∪v
    COVER_v=true
    for each vertex w ∈N(v)&&COVER_w=false do
      if(C_KC(w)==C_KC(v))
         COVER_w=true
      else
         C_nc(w)=C_nc(w)-C_nc(v)
    end for
end while
return S
```

The details of the *NCCDH* are described in the following steps.

Step 1 (lines 1-6): Initialize the seed sets S to null. For each node $u \in V$, the *nc* index is calculated according to formulas (1) and (2), and its status is initially set to uncovered.

Step 2 (lines 8-10): Select the node v with an uncovered status that has the largest *nc* index. This node will be selected as a seed and will not participate in subsequent rounds.

Step 3 (lines 11-16): Update each node w with the uncovered status that belongs to the neighbors of v. If w is located in the same shell layer as v, it will be directly covered. When w is outside the shell, the *nc* value of w will be discounted.

Step 4: Repeat steps 2 and 3 until k seeds are selected.

3.2 Sample analysis

To better illustrate the *NCCDH* algorithm, **Fig. 2** presents a detailed process to select the top 3 seeds in a sample network. In the first round of choosing the first seed, the initial status of each node is *false*, indicating that the node is not covered. The *NCCDH* will choose node A that has the largest *nc* ($C_{nc}(A)=12$) as the first seed and will cover the neighbors (B, C, and D) within the same shell layer of node A. In addition, the *nc* value of neighbor E and M outside the shell will be discounted from 7 to 4 and from 8 to 5, respectively. In the next round, because the node with the largest *nc* and uncovered status is H, node H will be chosen as the second seed. The remaining selections are performed in the same manner, and node M will be selected as the third seed. Then, the seed set is $S=\{A, H, M\}$.

Distinct from the *NCCDH*, the *MCC* algorithm will directly cover all neighbors of the seed after randomly obtaining one seed from the high-order shell layer. For example, the *MCC* will randomly select any node from the 3-shell *(A~D)* as the first seed and then cover all the neighbors of the seed and choose the second seed from the 2-shell. The *Degree* algorithm selects node M with the largest degree as the first seed, and the subsequent seeds are randomly selected from A, B, and H. Selecting nodes A and B as the 2nd and 3rd seed, respectively, gives highly coarse-grained results and overlapping spheres of influence. According to the above discussion, multiple candidates can be selected as the seed in each turn. Therefore, the *MCC* and *Degree* algorithms are unable to distinguish the differences between nodes at a fine-grained level.

The *CCA* algorithm selects the node located in the high-order shell with the largest degree as the initial seed. As shown in **Fig. 2**, the *CCA* will select nodes A or B with the largest degree in the 3-shell as the first seed. An assumption of the process is that node A will be selected as the first seed, nodes B, C, D, and E will be covered and the algorithm will choose node G, which is located in the 2-shell and has the largest degree, as the second seed. If the infection probability is β in the *SIR* model, where the immune probability is $\gamma=1$, and A is the first seed, then the spreading influence of selecting G as the second seed is calculated as $F_G(t)=1+3\beta+4\beta^2$ ($t\geq 2$). However, the spreading influence of H obtained by the *NCCDH* as the second seed is calculated as $F_H(t)=1+5\beta+2\beta^2$ ($t\geq 2$). Similarly, the propagation effect of the third seed (node S) obtained by the *CCA* is also less than that of the *NCCDH* (node M), where the spreading influence of S is $F_S(t)=1+2\beta+3\beta^2$ ($t\geq 2$) and the spreading influence of M is $F_M(t)=1+5\beta$($t\geq 2$). The spreading ability of the seeds obtained by the *CCA* is less than that of the *NCCDH*. In other words, the *NCCDH* can distinguish the role of the nodes to locate the seeds more accurately while reasonably evaluating the nodes with large degrees in the periphery of the network.

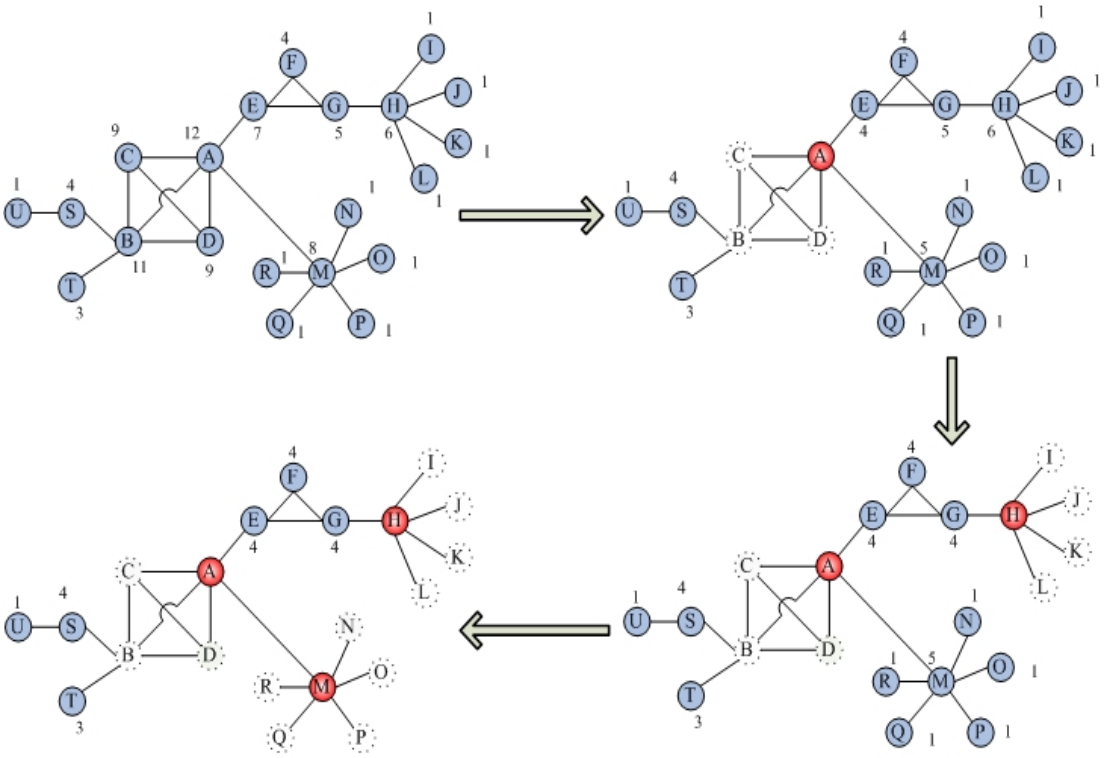


Fig. 2. Toy network for selecting the top 3 seeds with the *NCCDH*.

4. Evaluation and analysis

4.1 Network datasets

In the experiment, we use three real network datasets to evaluate the performance of the *NCCDH*. *Hamsterster friendships* [32] defines the friendships and family links between users of the website "www.hamsterster.com". The *Ca-GrQc* [33] is a collaboration network from the e-print arXiv and covers scientific collaborations between authors of papers submitted to the General Relativity and Quantum Cosmology category. The data cover papers published in the period from January 1993 to April 2003. Newman's *COND-MAT* [34] is the co-authorship network based on preprints posted to the Condensed Matter section of the e-print arXiv archived between 1995 and 1999. The topological features of these three networks are shown in **Table 1**, where n is the number of nodes, m is the number of edges, $<d>$ is the average degree, d_{max} is the maximum degree. T is the transitivity which measures the probability that the adjacent vertices of a vertex are connected, it is calculated as follows:

$$T = \frac{1}{n}\sum_{i=1}^{n} T_i = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{d_i(d_i-1)}\sum_{j,k=1}^{n} a_{ij}a_{jk}a_{ki} \tag{5}$$

where a_{ij} are elements of the adjacency matrix, d_i is the degree of node i. A is the assortativity that measures the degree to which similar vertices tend to connect to each other, and kc_{max} is the maximum value of the *k-core* centrality.

Table 1. Topological features of the three networks.

Network	*n*	*m*	*<d>*	d_{max}	*T*	*A*	kc_{max}
Hamsterster friendships	2426	16630	13.7098	273	0.2313	0.0474	24
Ca-GrQc	5242	14484	5.5261	81	0.6298	0.6593	43
COND-MAT	16264	47594	5.8527	107	0.3596	0.1846	17

4.2 Experimental setup

The two basic propagation models are applied in the experiment as explained in Section 2.3. In the *SIR* model, we mainly report the results on a small infection probability of $\beta \in$[0.01,0.02,0.03,0.04,0.05,0.06]. Larger β values, such as β=0.1, are not considered because of the insensitivity of the model to the different methods proposed in [23]. The immune probability of the *SIR* model is set to 1 (γ=1), which means that the *Infected* nodes will be recovered immediately after infecting their neighbors with a status of *Susceptible*. Under this circumstance, the *SIR* model is equivalent to the independent cascade model where any active node *i* has only one chance to infect its inactive neighbor *j*. Whether the infection is successful or not, node *i* will not infect *j* in subsequent steps. This model is widely used for cascading problems, such as traffic jams, financial systems and retweeting behaviors. The experiment runs 10,000 times to obtain the average result. The running platform is an Intel Core i3-2348M with 4 G RAM, and the programming environment is R 3.1-win.

The performance of the algorithm is analyzed from the following aspects. (*a*) For each algorithm, the spreading influence with different seed set sizes ranging from 1 to 50 is compared. (*b*) The impact of the infection probability β on the information propagation is analyzed while selecting *k*=50 seeds. (*c*) The spreading influences of different algorithms in the *SIR* and *SI* models are evaluated, where the *SIR* model is used to verify the affected scale and the *SI* model is used to verify the propagation velocity. (*d*) The running time for the seed selection is determined. Because of the high time complexity, the greedy algorithm is not considered in the experiment. The algorithms used here are shown in **Table 2**.

Table 2. Experimental algorithms.

Algorithm	*Description*
Degree	Selecting nodes with the top-*k* largest degrees as seeds.
k-shell	Selecting nodes with the top-*k* largest *k-core* centrality as seeds.
MCC	Covering all the neighbors of a selected seed by the maximum *k-core* centrality.
PageRank	An algorithm used by Google Search to rank websites in their search engine. The damping factor is 0.85 and the converging threshold is 0.001.
DDH	Discounting the degrees of the seed's neighbors.
NCCDH	Covering the neighbors inside the shell layer and discounting the neighbors outside the shell layer for a seed obtained by the neighborhood coreness.

4.3 Experiment results

(*a*) Spreading influence with different numbers of seeds

Fig. 3 shows the spreading influence with the number of seeds in the *SIR* model. **Fig. 3 *(a)***,

(b), and ***(c)*** shows that the *DDH* can achieve the best results when the probability of infection is smaller (β=0.01). Therefore, the performance of the *DDH* is better than that of the *Degree*, *PageRank*, *NCCDH*, *MCC* and *k-shell* algorithms. Although the performance of the *NCCDH* has been greatly improved as shown in **Fig. 3*(b)(c)*** and outperforms the *Degree*, *PageRank*, and *MCC* algorithms, the spreading scale of the *NCCDH* is closer to that of the *DDH* because when the probability of infection is smaller, the node can spread only within a limited depth range. Therefore, the spreading sphere of the nodes with larger degrees will be wider. Because the *DDH* considers both the degree and the decentralization for the seed selection, the method has the best spreading results when the probability of infection is smaller. The *NCCDH* considers the seed's position and decentralization; however, the node in the core of the network does not always have the largest degree. Therefore, the advantage is not obvious when the infection probability is smaller. Compared with **Fig. 3(*a*),** the spreading ability of the *NCCDH* in **Fig. 3*(b)*** and ***(c)*** is better than the *Degree*, *PageRank*, *MCC* and other algorithms. This phenomenon can be explained by the greater transitivity of the *Ca-GrQc* and *COND-MAT* networks (shown in **Table 1**), which is beneficial to the spreading influence for the nodes located in the core of the network when the infection probability is smaller.

When β is defined as 0.03 and 0.06, the *NCCDH* outperforms all other algorithms. In addition, the *MCC* algorithm, which randomly chooses the seeds in the high-order shell, even achieves good performance in the *Hamsterster friendships* and *Ca-GrQc*. **Fig. 3 *(d)***, ***(e)***, ***(f)***, ***(g)***, ***(h)***, and ***(i)*** shows that the advantage of the *NCCDH* is more obvious with increases in the value of *k*, which can be explained by the greater impact of the decentralized seeds in the core of the network on the spreading influence when the infection probability is increased compared with the impact of the nodes that have a large degree but may be located in the periphery. Moreover, when *k* increases, the growth rate of the spreading influence by the *k-shell* tends to be slower, which can be explained by the clustering of the seeds obtained by the *k-shell* without considering the overlapping regions.

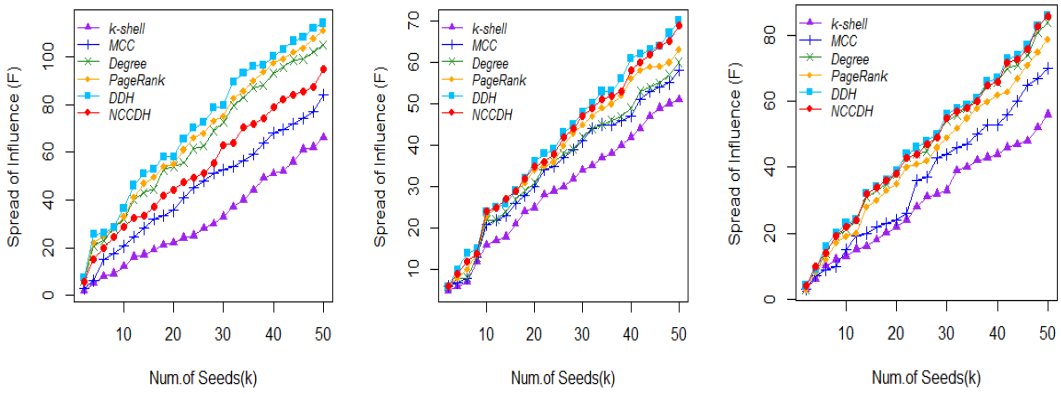
(b) Spreading influence with different infection probabilities

Almost all nodes can be infected when β is set to a large value; in addition, the role of the seeds is no longer important because the final affected scale is independent of spreaders' location. To distinguish the spreading results, the infection probability β is set to [0.01, 0.06]. **Fig. 4 *(a)***, ***(b)***, and ***(c)*** show the spreading results of different probabilities in the *SIR* model with a fixed seed number (k=50). **Fig. 4*(a)***, ***(b)***, and ***(c)*** show that when $\beta \geq 0.03$, the advantage of the *NCCDH* is more obvious than all the other algorithms, especially in the *Ca-GrQc* and *COND-MAT*. For example, limited spreading is promoted by applying the *k-shell* decomposition on the *Ca-GrQc and COND-MAT* because these seeds are mostly in close proximity to one another. When β is too small ($\beta \leq 0.02$), information can spread only at a finite depth regardless of how the seeds are selected; therefore, the *DDH* algorithm, which is based on the degree of centrality, has a slight advantage. However, the *NCCDH* considers the position of the node and the neighborhood information. As a result, the spreading influence of the *NCCDH* is always better than that of the *MCC* and *k-shell*. Therefore, the *NCCDH* has a better spreading influence with respect to other benchmark algorithms when the infection probability increases.

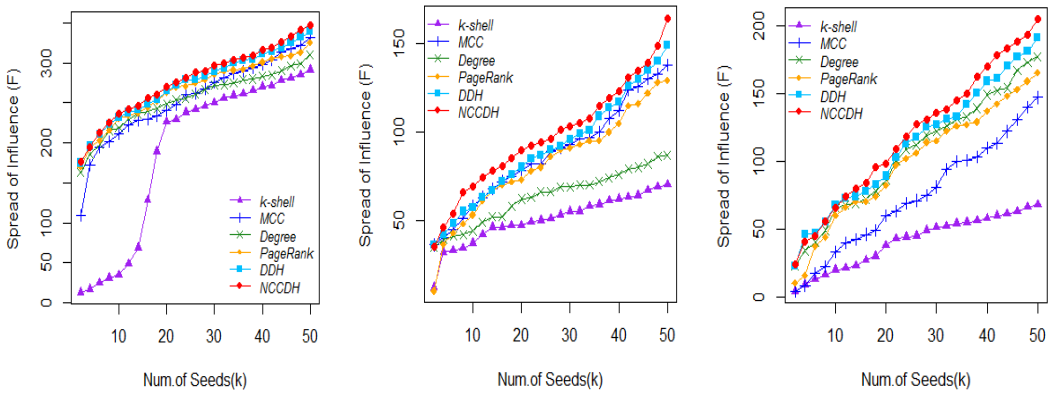
(c) Spreading influence in the *SIR* and *SI* models

Fig. 5 shows the spreading influence at time *t* in the *SIR* model, with k=50 and β=0.06. **Fig. 5*(a)***, ***(b)***, and ***(c)*** show that the seeds obtained by the *NCCDH* can generate a larger scale of influence than that of the other algorithms. For example, in terms of the spreading influence,

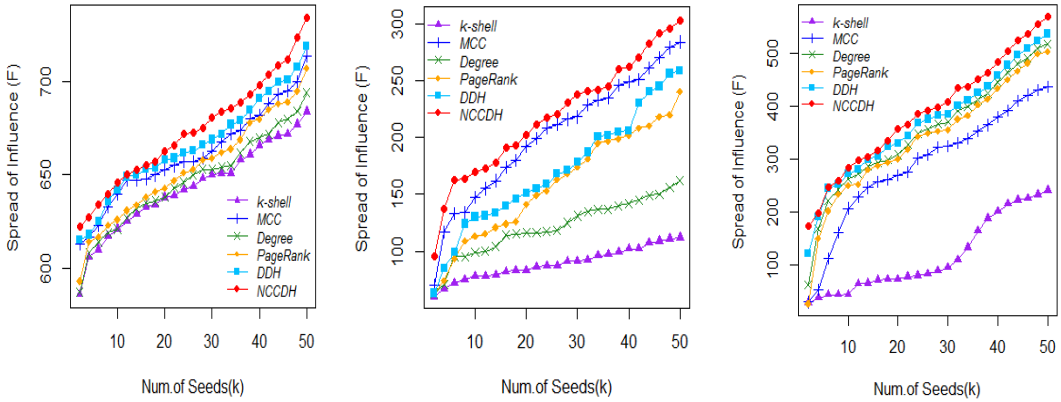
the *NCCDH* outperforms the *DDH*, *PageRank*, *Degree* and *k-shell* by 3.98%, 5.86%, 7.79%, and 12.44% in *Hamsterster friendships*, respectively, and by 8.63%, 17.87%, 28.84%, and 47.3% in the *Ca-GrQc*, respectively. Because the *SI* model can eventually infect all nodes (steady state), the model is often used to evaluate the spreading speed of different algorithms. **Fig. 6*(a)* and *(b)*** represent the spreading influence at time *t* in the *SI* model, with *k*=50 and *β*=0.06. The experimental results are given here for the *Ca-GrQc* and *COND-MAT* with a larger scale. The proposed method is always better than the other methods for the average spreading influence at each step. For example, in the *Ca-GrQc*, the *NCCDH* infected 90.31% of nodes at *t*=10 and achieved the steady state at *t*=16, whereas the *DDH* and *Degree* infected 83.45% and 76.36% of nodes at *t*=10, respectively, and reached the steady state at *t*=17 and *t*=18, respectively. The propagation speed of *NCCDH* is faster than that of the other methods in the *SI* model. In other words, less time is required to achieve the same spreading scale.



(a) Hamsterster friendships (β=0.01) *(b) Ca-GrQc (β=0.01)* *(c) COND-MAT (β=0.01)*

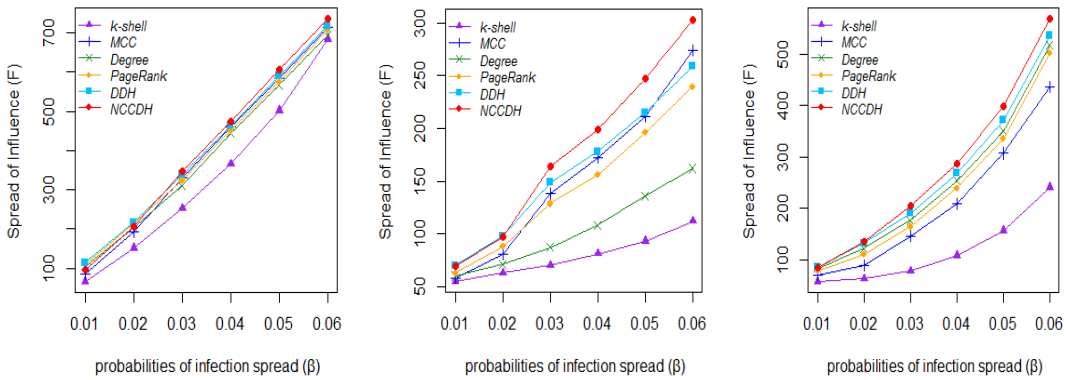


(d) Hamsterster friendships (β=0.03) *(e) Ca-GrQc (β=0.03)* *(f) COND-MAT (β=0.03)*



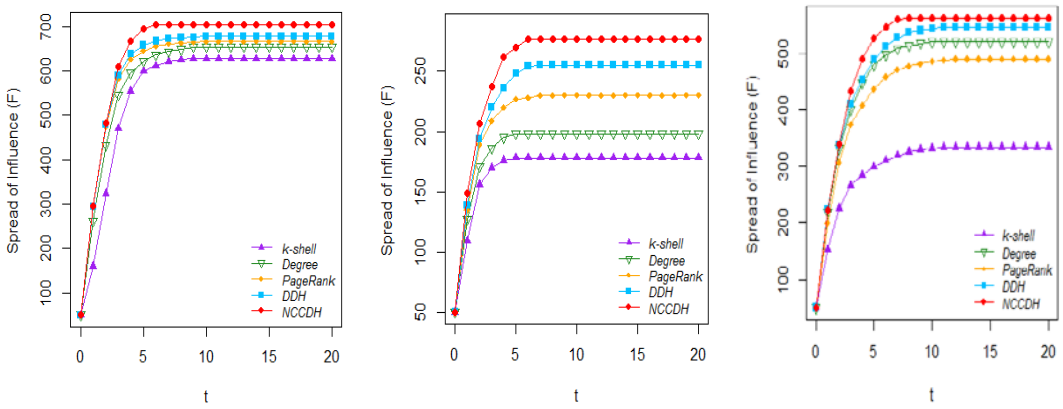
(g) Hamsterster friendships (β=0.06) *(h) Ca-GrQc (β=0.06)* *(i) COND-MAT (β=0.06)*

Fig. 3. Spreading influence F in the *SIR* with different numbers of seeds $k\in[1,50]$ and $\beta\in\{0.01,0.03,0.06\}$. (*a*–*c*) *Hamsterster friendships*; (*d*–*f*) *Ca-GrQc*; and (*g*–*i*) *COND-MAT*.



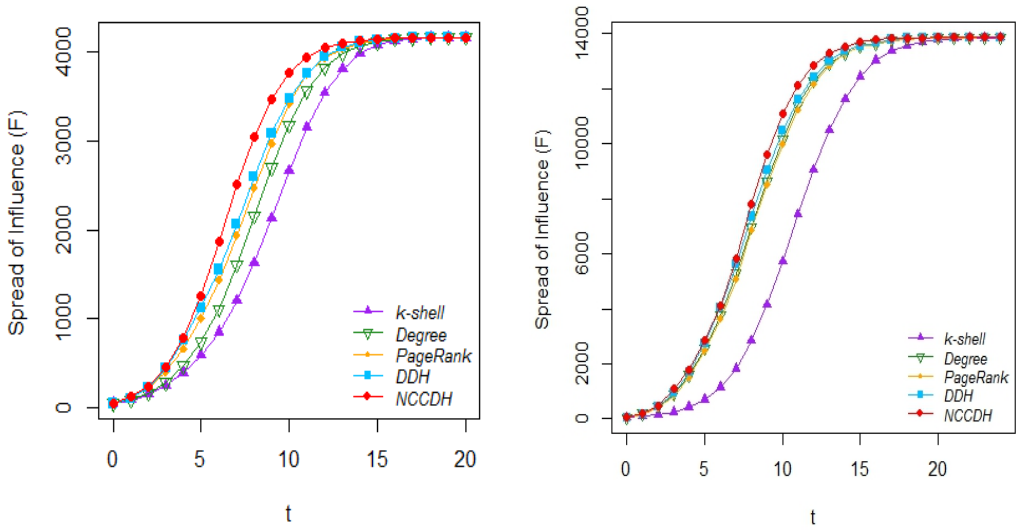
(a) Hamsterster friendships *(b) Ca-GrQc* *(c) COND-MAT*

Fig. 4. Spreading influence F in the *SIR* with different infection probabilities $\beta\in\{0.01,0.02,0.03,0.04,0.05,0.06\}$, with k=50. *(a) Hamsterster friendships; (b) Ca-GrQc; and (c) COND-MAT.*



(a) Hamsterster friendships *(b) Ca-GrQc* *(c) COND-MAT*

Fig. 5. Spreading influence *F* in the *SIR*, where β=0.06 and *k*=50. *(a) Hamsterster friendships; (b) Ca-GrQc; and (c) COND-MAT.*



(a) Ca-GrQc *(b) COND-MAT*

Fig. 6. Spreading influence *F* in the *SI* model where β=0.06 and *k*=50. *(a) Ca-GrQc and (b) COND-MAT.*

(*d*) Running time

K-core decomposition can be applied to large-scale networks because of its low computational complexity. Therefore, the *NCCDH* algorithm based on the *nc* obtained by the *k*-core decomposition also has a superior computational efficiency. The computational time of the *NCCDH* is mainly composed of four parts: the time to initialize the *k-core* centrality for all

nodes, the time to calculate the *nc* for all nodes, the time to select a node with the largest *nc* value and uncovered status, and the time to cover or discount the neighbors of a seed.

First, the shell indices of all nodes are calculated by the *k-core* decomposition, where the time complexity is $O(m)$ and m is the number of edges in a network. Second, the *nc* index of each node is calculated according to the *k-core* positions of the neighbors, where the time complexity is $O(<d>*n)=O(m)$, $<d>$ is the average degree of the network and n is the number of nodes. Third, the node that has the largest *nc* value and uncovered status in each round will be selected as a seed, where the time complexity is $O(n)$. Finally, the neighbors of the node that has been selected as a seed will be covered or discounted where the time complexity is $O(<d>)=O(m/n)$. Therefore, to select k seeds with k times in steps 3 and 4, the total computational complexity of the *NCCDH* is $O(m)+O(m)+O(k*n)+O(k*m/n)$. If $k \ll n$ and the networks are sparse, the time complexity of the *NCCDH* is approximately $O(n)$.

The running time for the seed selection in three real networks is shown in **Table 3**. This table shows that because the *Degree* algorithm considers only the local properties, this method achieves the highest computational efficiency. The computational efficiency of the *NCCDH* algorithm is better than that of the *DDH* and *PageRank* algorithms because once a node is selected as the seed, the neighbors within the same shell layer of this node will be covered. In a subsequent selection, those covered nodes will be directly ignored to reduce the computational complexity. The computational efficiency of the *DDH* and *NCCDH* in the *Ca-GrQc* is better than that of the *Hamsterster friendships*, which can be interpreted as the lower average degree $<d>$ in the *Ca-GrQc* (shown in **Table 1**). Therefore, it is beneficial to reduce the complexity of the discount calculation.

Table 3. Running time for selecting seeds in the three networks (seconds).

Network	***Algorithm***	***k=10***	***k=20***	***k=30***	***k=40***	***k=50***
Hamsterster friendships	*Degree*	0.2340	0.2340	0.2340	0.2340	0.2340
	k-shell	0.2360	0.2360	0.2360	0.2360	0.2370
	MCC	1.5600	2.2308	2.9172	3.1668	3.4994
	PageRank	60.946	60.946	60.946	60.983	60.983
	DDH	4.6956	7.0824	9.1260	11.4036	12.6984
	NCCDH	3.7128	5.3820	6.6300	7.7532	8.7048
Ca-GrQc	*Degree*	0.2495	0.2495	0.2495	0.2496	0.2496
	k-shell	0.2496	0.2498	0.2652	0.2652	0.2652
	MCC	1.1232	1.4820	1.6536	1.9500	2.1684
	PageRank	40.0862	40.0863	40.0879	40.0879	40.0879
	DDH	2.5428	3.8532	4.9296	5.6940	6.6924
	NCCDH	2.0904	2.7612	3.3696	3.9624	4.4616
COND-MAT	*Degree*	0.2496	0.2496	0.2651	0.2652	0.2652
	k-shell	0.2963	0.2963	0.2964	0.2964	0.2964
	MCC	2.0905	3.3852	4.5708	5.5536	6.5676
	PageRank	124.764	124.764	124.782	124.782	124.782
	DDH	5.5848	9.0168	12.2772	15.0696	18.8400
	NCCDH	5.4444	8.8608	11.7936	14.6172	17.2224

5. Conclusions

In this paper, we focused on limitations of the heuristic influence maximization algorithms in current social networks and proposed a *nc* cover and discount heuristic algorithm named "*NCCDH*". After choosing a node as the seed in each turn, the neighbors within the same shell

layer of this seed are covered. At the same time, the *nc* of the neighbors outside the shell will be discounted in the subsequent round. The *NCCDH* can not only prevent many nodes with overlapping spheres in the high-order shell from being selected as the seeds but also overcome the shortcoming in which nodes with a large degree in the periphery of the network are not given a reasonable assessment. The method can identify a set of influential and decentralized seeds at a more fine-grained level. The experimental results show that because of its increasing spreading probability, the *NCCDH* outperforms other benchmarks in terms of the affected scale and spreading speed under the *SIR* and *SI* models. More importantly, the time complexity of this approach is superior. Recent works have begun to study temporal networks; thus, identifying influential spreaders in a temporal network will become an important topic in related fields in the future. The network community structure will also have an important effect on the influence maximization. Extending our work to these fields is worthy of further study.

References

[1] L Y Lü, D B Chen, X L Ren, Q M Zhang, Y C Zhang and T Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol.650, pp.1-63, September, 2016. Article (CrossRef Link).

[2] H Li, J T Cui and J F Ma, "Social influence study in online networks: A three-level review," *Journal of Computer Science and Technology*, vol.30, no.1, pp.184-199, January, 2015. Article (CrossRef Link)

[3] P.S.Romualdo and V.Alessandro, "Immunization of complex networks," *Physical Review E*, vol.65, no.3, pp.036104, April, 2002. Article (CrossRef Link)

[4] J. Leskovec, L.A. Adamic and B.A. Huberman, "The dynamics of viral marketing," *ACM Transactions on the Web*, vol.1,no.1, pp.1-39, April, 2007. Article (CrossRef Link)

[5] A.E.Motter, "Cascade control and defense in complex networks," *Physical Review Letters*, vol.93, no.9, pp.98701, September, 2004. Article (CrossRef Link)

[6] W Chen, L.V.S. Lakshmanan and C. Castillo, "Information and influence propagation in social networks," *Synthesis Lectures on Data Management*, vol.5, no.4, pp.1-177, October, 2013. Article (CrossRef Link)

[7] F. Radicchi, "Who is the best player ever? A complex network analysis of the history of professional tennis," *PLoS ONE*, vol.6, no.2, pp.e17249, February, 2011. Article (CrossRef Link)

[8] Y B Zhou, L Lü and M Li, "Quantifying the influence of scientists and their publications: distinguishing between prestige and popularity," *New Journal of Physics*, vol.14, no.3, pp. 33033-33049, March, 2012. Article (CrossRef Link)

[9] L C Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol.1, no.3, pp.215-239, March, 1979. Article (CrossRef Link)

[10] G Sabidussi, "The centrality index of a graph," *Psychometrika*, vol.31, no.4, pp.581-603, April, 1966. Article (CrossRef Link)

[11] L Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol.18, no.1, pp.39-43, March, 1953. Article (CrossRef Link)

[12] M Kitsak, L K Gallos and S Havlin, "Identification of influential spreaders in complex networks," *Nature Physics*, vol.6, no.11, pp.888-893, August, 2010. Article (CrossRef Link)

[13] A Zeng and C J Zhang, "Ranking spreaders by decomposing complex networks," *Physics Letters A*, vol.377, no.14, pp.1031-1035, June, 2013. Article (CrossRef Link)

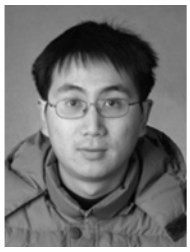
[14] J G Liu, Z M Ren and Q Guo, "Ranking the spreading influence in complex networks," *Physica A*, vol.392, no.18, pp.4154-4159, September, 2013. Article (CrossRef Link)

[15] S Pei, L Muchnik, J S Andrade, Z Zheng and H A Makse, "Searching for superspreaders of information in real-world social media," *Scientific Reports*, vol.4, pp.5547, July, 2014. Article (CrossRef Link)

[16] J Bae and S Kim, "Identifying and ranking influential spreaders in complex networks by neighborhood coreness," *Physica A*, vol.395, no.4, pp.549-559, February, 2014. Article (CrossRef Link)
[17] Y Liu, M Tang, T Zhou and Y Do, "Improving the accuracy of the k-shell method by removing redundant links: from a perspective of spreading dynamics," *Scientific Reports*, vol.5, pp.13172, May, 2015. Article (CrossRef Link)
[18] Y Liu, M Tang, T Zhou and Y Do, "Core-like groups result in invalidation of identifying super-spreader by k-shell decomposition," *Scientific Reports*. vol.5, pp.9602, May, 2015. Article (CrossRef Link)
[19] Y H Fu, C Y Huang and C T Sun, "Using global diversity and local topology features to identify influential network spreaders," *Physica A*, vol.433, no.9, pp.344-355, September, 2015. Article (CrossRef Link)
[20] PGV Naranjo, M Shojafar, H Mostafaei ,Z Pooranian and E Baccarelli, "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *Journal of Supercomputing*, vol.73, no.2, pp.733-755, February, 2017. Article (CrossRef Link)
[21] Z Pooranian, M Shojafar, JH Abawajy and A Abraham, "An efficient meta-heuristic algorithm for grid computing," *Journal of Combinatorial Optimization*, vol.30, no.3, pp.413-434, October, 2015. Article (CrossRef Link)
[22] Z Pooranian, N Nikmehr, S Najafi-Ravadanegh, H Mahdin and J Abawajy, "Economical and Environmental Operation of Smart Networked Microgrids under Uncertainties Using NSGA-II," in *Proc. of the 24th International Conference on Software, Telecommunications and Computer Networks*, pp.1-7, September 22 - 24, 2016. Article (CrossRef Link)
[23] D Kempe, J Kleinberg and E Tardos, "Maximizing the spread of influence through a social network." in *Proc. of the 9th ACM Conference on Knowledge Discovery and Data Mining*, pp.137-146, August 24 - 27, 2003. Article (CrossRef Link)
[24] Z K Zhang, C Liu, X X Zhan, X Lu, C X Zhang, Y C Zhang, "Dynamics of information diffusion and its applications on complex networks," *Physics Reports*, vol.651, pp.1-34, September, 2016. Article (CrossRef Link)
[25] J Leskovec, A Krause, C Guestrin, C Faloutsos, J VanBriesen, and N Glance, "Cost-effective outbreak detection in networks," in *Proc. of the 13th ACM Conference on Knowledge Discovery and Data Mining*, pp.420-429, August 12 - 15, 2007. Article (CrossRef Link)
[26] W Chen, Y Wang and S Yang, "Efficient influence maximization in social networks," in *Proc. of the 15th ACM Conference on Knowledge Discovery and Data Mining*, pp.199-208, June 28 - July 01, 2009. Article (CrossRef Link)
[27] P SanKar, S Kundu, and CA Murthy, "Centrality Measures, Upper Bound, and Influence Maximization in Large Scale Directed Social Networks," *Fundamenta Informaticae*, vol.130, no.3, pp.317–342, July, 2014. Article (CrossRef Link)
[28] H Kim, K Beznosov and E Yoneki, "A study on the influential neighbors to maximize information diffusion in online social networks," *Computational Social Networks*, vol.2, no.1, pp.1-15, February, 2015. Article (CrossRef Link)
[29] Y Liu, M Tang and T Zhou, "Identify influential spreaders in complex networks, the role of neighborhood," *Physica A*, vol.452, no.6, pp.289-298, June, 2016. Article (CrossRef Link)
[30] J X Zhang, D B Chen, Q Dong and Z D Zhao, "Identifying a set of influential spreaders in complex networks," *Scientific Reports*, vol.6, pp.27823, June, 2016. Article (CrossRef Link)
[31] J X Cao, D Dong, X Shun, X Zheng, B Liu and J Z Luo, "A k-core based Algorithm for Influence Maximization in Social Networks," *Chinese Journal of Computers*, vol.38, no.2, pp.238-248, February, 2015. Article (CrossRef Link)
[32] R A Rossi and N K Ahmed, "An Interactive Data Repository with Visual Analytics," *ACM SIGKDD Explorations Newsletter*, vol.17, no.2, pp.37-41, February, 2016. Article (CrossRef Link)

[33] J Leskovec, J Kleinberg and C Faloutsos, "Graph Evolution: Densification and Shrinking Diameters," *ACM Transactions on Knowledge Discovery from Data*,vol.1, no.1, pp.1-41, March, 2007. Article (CrossRef Link)

[34] M E Newman, "The structure of scientific collaboration networks," in *Proc. of the National Academy of Sciences*, vol.98, no.2, pp.404-409, February, 2001. Article (CrossRef Link)



Xiong Yang received his B.E degree and M.E degree in Information Engineering from Nanjing University of Information Science and Technology in 2003 and 2006, respectively. He is a PhD candidate at the Zhejiang University of Technology ,China. He had been a visiting scholar at University of Hertfordshire in UK from 2016 to 2017. Meanwhile, He is a lecturer working in the Changzhou Institute of Technology, China. His research interests include complex networks and data mining.

DeCai Huang received his Ph.D. degree from the ChongQin University ,China. He is currently an professor in School of Computer Science and Technology at Zhejiang University of Technology, China. He is interested in data Mining and database. He has been either an author or a co-author of over 70 papers in academic journals (international or Chinese) and high-profile international conferences held by the IEEE organization.

ZiKe Zhang received his Ph.D. degree from the University of Fribourg, Switzerland. He has been a Postdoc in the Alibaba Research Center for Complexity Sciences since 2014 to 2016. He is currently a professor in Hangzhou Normal University, China. His research interests include information propagation, complex networks, and recommendation system.