Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3099

KEY NODE IDENTIFICATION ALGORITHM BASED ON LOCAL SEMI GLOBAL TRIANGLE CALCULATION

HanYi Yang[1*], YiJia He[2]
[1]*College of Cyber Security, Tarim University, Alar 843300, Xinjiang, China.*
[2]*College of Foreign Languages, Tarim University, Alar 843300, Xinjiang, China.*
Corresponding Author: HanYi Yang, Email: xzmu_yhy@qq.com

Abstract: Aiming at the challenges of low identification accuracy and slow computation time in existing key node identification algorithms for complex networks, the paper proposes a key node identification algorithm based on local semi global triangular computation (LSTC). First, inspired by the structural stability of triangles in the physical world, the triangular patterns of nodes in complex networks and their importance are defined. Second, drawing on the third-order partition theory which highlights strong connections between a node and its third-order neighbors, the algorithm incorporates the influence of a node's local third-order neighbors when evaluating its importance. To validate the experimental performance of the proposed algorithm, the LSTC algorithm is compared with eight other algorithms of the same type using both the Susceptible–Infected–Recovered (SIR) model and the Linear Threshold (LT) model. Experimental results demonstrate that the proposed algorithm achieves the highest overall performance.
Keywords: Complex network; Influential spreaders; Spreading ability; SIR epidemic model

1 INTRODUCTION

Network communication is a natural phenomenon in many fields of life[1], such as pandemic, disease transmission, information transmission, etc. We must adjust or maximize the communication process according to social interests and requirements. Influential communicators play a key role in optimizing or managing the impact of the communication process. The most influential extender is an important node in the network, which acts as the maximizer or controller of the extender process. In the real world, influential communicators have many applications in various fields, such as spreading information on social networks, and controlling rumors or epidemics in the system. In order to identify influential communicators from the network, researchers have proposed various centralized methods. The central approach is also used in other areas. For example, measure the social impact capacity of scientists in the cooperative network to determine the important economic pillars in the economic network, investigate the impact and timeliness of journals in the scientific citation network, and find communities. In this article, we focus on identifying influential communicators from the network. The schematic diagram of a complex network is shown in Figure 1:

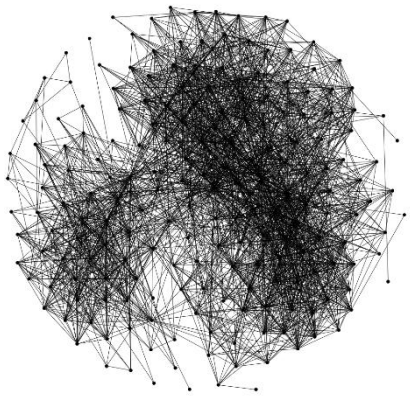

Figure 1 Complex Network

The identification of key nodes in complex networks is reflected in various fields. In the public health epidemic [2], identifying key individuals will help optimize vaccination strategies to contain the outbreak of the epidemic [3]. In social media and digital marketing, identifying well connected users can maximize the scope and impact of information activities and spread content. Similarly, in network security and error information control, locating the central node allows to contain rumors or malicious attacks in the network system. In addition, the centrality based approach also helps to quantify the scientific impact in academic cooperation networks, identify systemically important entities in financial networks, evaluate the reputation of journals in citation networks, and detect community structures in social and biological networks.

Although researchers have designed various centrality measures to sort and select the importance of complex network nodes, many measures are limited by their dependence on specific structural attributes or computational constraints. In order to solve the problem, this paper pays special attention to the identification of influential communicators in complex networks. We systematically reviewed and classified the existing centralized methods, emphasized their advantages and limitations, and stimulated the demand for more robust and efficient methods that can adapt to different network topologies and dynamic contexts.

Among various selection algorithms, the most widely used metric is usually based on the topological position of nodes or some aspects of extended dynamics, such as random walking, shortest path distance and information content. These centrality methods can be roughly divided into four types: local centrality, global centrality, semi global centrality and mixed centrality.

2 RELATED WORK

2.1 Basic Algorithm

The key node identification algorithm can be applied to solve various real-world problems. For example, in a water supply network, pollutants in the entire system can be monitored by identifying key locations and deploying sensors at these points. The core challenge of such algorithms is how to effectively and accurately identify the most influential nodes in the network. In recent years, researchers have proposed a series of key node identification methods, which are summarized as follows:

(1) Degree

The most basic topological property in the network is the degree of the node. The algorithm based on degree centrality (Degree algorithm) [4] believes that the greater the degree of the node, the higher the influence in the network, that is, the more direct neighbors of the node, the more important the node is. The method for calculating node degree centrality is defined as:

$$DC_v = \frac{|\Gamma_v|}{N-1} \tag{1}$$

Where, Γ_v is the set of direct neighbors of a node v, $|\Gamma_v|$ is the module of the set of direct neighbors of a node, N is the total number of nodes in the network, and $N-1$ is the number of other nodes in the network except nodes.

(2) H-index

H-index [5] is a widely used measurement standard, which is used to evaluate the scientific research output of researchers and the influence of papers. Early measurement of academic influence usually relied only on isolated indicators, such as the total number of publications or the number of citations of each paper. However, the H index achieves a more balanced assessment by taking into account the number of authors' publications and the corresponding number of citations. This dual consideration enables a more accurate and robust reflection of a scholar's academic influence. H is defined as an operator, which only acts on the finite set of real numbers $(x_1, x_2, \cdots, x_n)$, so H-index is defined as:

$$y = H(x_1, x_2, \cdots, x_n) \tag{2}$$

Where, $y > 0$ and y is the largest integer, the H operator makes that there are at least y elements in $(x_1, x_2, \cdots, x_n)$, and each element is not less than y. In other words, for a scholar with n papers, $(x_1, x_2, \cdots, x_n)$ is the number of citations of the papers, and $H(x_1, x_2, \cdots, x_n)$ is the H-index value of the scholar.

(3)K-shell

The K-shell algorithm was first proposed by Kitsak et al [6]. and used to find influential nodes in the network. The algorithm believes that the closer the node is to the center of the network, the greater its importance and influence. The specific process of K-shell algorithm is as follows:

(a) Count the degree of each node in the network and record the minimum degree $k_{\min}$.

(b) The k-shell decomposition process iteratively prunes nodes with degree $k_{\min}$ from the network. After each removal phase, the degrees of remaining nodes are updated, and the procedure repeats for the new graph structure. Each pruned set of nodes is assigned the current k-shell value. Nodes removed in later iterations—those persisting in more densely connected core regions—receive higher k-shell values, reflecting their greater structural centrality within the network.

(c) Repeat (a) and (b) until there are no connected nodes in the network, that is, each node has a k-shell value.

(4) ISK

Based on the K-shell algorithm, Wang et al[7]. introduced the ISK algorithm by incorporating the concept of information entropy—also referred to as Shannon entropy, which serves as a quantitative measure of information. The IKS algorithm utilizes the magnitude of a node's information entropy to differentiate nodes that share the same k-shell value, thereby enabling a more refined and accurate ranking of node influence within the same shell layer. First, the local importance of node i is defined:

$$I_i = k_i \Big/ \sum_{j=1}^{N} k_j \tag{3}$$

Where, k_i is the degree of the node, and j is the direct neighbor of the node i. Secondly, the method to calculate the node information entropy is as follows:

$$e_i = -\sum_{j\in\Gamma(i)} I_j \cdot \ln I_j \tag{4}$$

(5) MCDE

Based on the idea of hierarchical division of nodes in K-shell algorithm, many scholars have proposed various improved algorithms based on K-shell. Among them, Sheikhahmadi et al. [8]. This paper introduces the MCDE algorithm, which integrates the degree, k-shell value and information entropy of nodes into the weighted comprehensive measure. This multidimensional approach aims to capture node impacts more comprehensively. It is worth noting that the calculation of information entropy in MCDE algorithm is slightly different from that used in IKS algorithm. The specific formula of information entropy in MCDE is defined as follows:

$$Entropy(v) = -\sum_{i=0}^{Core_{\max}} (p_i \cdot \log_2 p_i) \tag{5}$$

Where, $p(i)$ is the ratio of the cumulative sum of the k-shell values of the direct neighbors of the node to the node degree, and the calculation formula is as follows:

$$p_i = \frac{\sum_{j\in\Gamma(i)}^{N} \text{k-shell(j)}}{k(i)} \tag{6}$$

Based on the above considerations, the MCDE algorithm is formally defined as follows:

$$MCDE(v) = \alpha \cdot \text{k-shell}(v) + \beta \cdot k(v) + \gamma \cdot Entropy(v) \tag{7}$$

Where, α , β , and γ are respectively k-shell values, node degrees, and weights of information entropy.

(6) ECRM

Based on the principle that the node importance in clustering algorithm can be calculated through the interaction between direct neighbors, Zareie et al. [9]. ECRM algorithm is proposed based on clustering algorithm. This method goes beyond simply calculating the shared neighbors between nodes and their direct connections; It also integrates the structural commonalities between them. Specifically, the algorithm makes use of the similarity and correlation between the connection modes of the connecting node and its immediate neighbors.

(7) VoteRank

In 2016, Zhang et al [10]. pioneered the application of a voting strategy to identify influential nodes by proposing the VoteRank algorithm. This method initializes each node in the network with a certain voting capacity, allowing it to cast votes in favor of its direct neighbors. During each iteration, the node accumulating the highest number of votes is identified as the most influential node in that round. The detailed procedure of the VoteRank algorithm is outlined as follows:

(a) Each node in the network is assigned a tuple (voting score, voting capacity), that is, (s_u, va_u) , and initialized to $(s_u, va_u) = (0,1)$.

(b) It is stipulated that each node can vote for its immediate neighbors according to its own voting capacity. The voting score of a node is the sum of the voting capacity of each neighbor. The voting score is calculated as:

$$s_u = \sum_{v\in\Gamma_u} va_v \tag{8}$$

(c) The node with the highest voting score in the current round is selected as the influential node. Once selected, this node is excluded from participating in any subsequent voting rounds.

(d) The voting capacity of the direct neighbors of the selected influential node is reduced by a factor equal to the reciprocal of the network's average degree $< k >$, as defined by the following expression:

$$va_u = va_u - \frac{1}{< k >} \tag{9}$$

(e) Repeat step (b), (c) and (d) until the specified L influence nodes are selected, where $L < N$.

(8) EnRenew

Based on the VoteRank algorithm, Guo et al. [11]. The EnRenew algorithm was proposed, which combines node information entropy to enhance the original method. This method effectively addresses the lack of discrimination in initial voting capacity allocation and subsequent voting decay processes in VoteRank. This algorithm utilizes node information entropy to initialize different voting abilities and weakening factors for different nodes, thereby achieving a more refined and adaptive node influence evaluation mechanism.

The EnRenew algorithm has been innovated in the initialization and weakening stages of the voting process. It uses information entropy to determine the initial voting ability and applies different attenuation factors during the weakening stage. Compared with the VoteRank algorithm that uses unified voting parameters, this improved method has achieved significant improvements in algorithm performance and recognition accuracy.

2.2 Analysis Summary

The above eight key node recognition algorithms have limitations in accuracy and computational efficiency, but the VoteRank algorithm provides an innovative solution by introducing some methods from society into complex networks. Similarly, this article also draws on the stability of triangular structures in the real world to construct a node neighbor triangular pattern in complex networks. At the same time, it draws on the strong connection between nodes and their

three boundary neighbors in the theory of three-degree separation, and comprehensively considers the local importance of nodes. The next section will provide a detailed introduction to our method.

3 PROPOSED ALGORITHM

3.1 Basic Concept

In large-scale real complex networks, a small number of key nodes often have a significant impact on the overall system behavior. Therefore, accurately identifying key nodes in a network plays an important role in the overall inference of complex networks. In order to improve the accuracy and efficiency of key node recognition, this paper proposes a key node recognition algorithm based on local semi global triangular model calculation. Firstly, inspired by the inherent stability of triangular structures in the real world, the concept of triangular patterns was introduced into complex networks, and the triangles formed between nodes and their neighbors were calculated. Secondly, based on the strong connection between nodes and their third-order neighbors in the theory of three degree segmentation, the importance of nodes is fully considered by taking into account their third-order neighborhoods.

3.2 Triangle Mode Calculation

The size of network density is closely related to the number of triangles in the network. The more triangles the network has, the greater the network density; The smaller the number of triangles, the smaller the network density. For a complex network $G=(V,E)$ with no right and no direction, its network density is calculated as follows:

$$D=\frac{E}{|V|*(|V|-1)/2} \tag{10}$$

Where E is all the edges in the network, and V is the collection of nodes in the network.
For a complete network or a completely dense network, the density is $D=1$. In a completely dense network, the total number of triangles is calculated as follows:

$$^{|V|}C_{R=3}=\frac{|V|!}{3!(|V|-3)!} \tag{11}$$

Where, $R=3$ indicates that the triangle vertex contains three nodes. The specific calculation is shown in Figure 2:

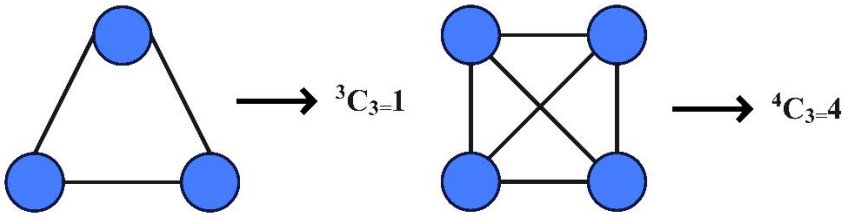

Figure 2 Triangle Calculation

Figure 2 illustrates the triangular counting principle in fully connected networks. The left panel depicts a 3-node fully connected network, which contains exactly one triangle. The right panel shows a 4-node fully connected network (K4), where the total number of distinct triangles amounts to four.
The four node fully connected network in the right figure is gradually removed. The calculation process of network density and triangle number in the removal process is shown in Figure 3:

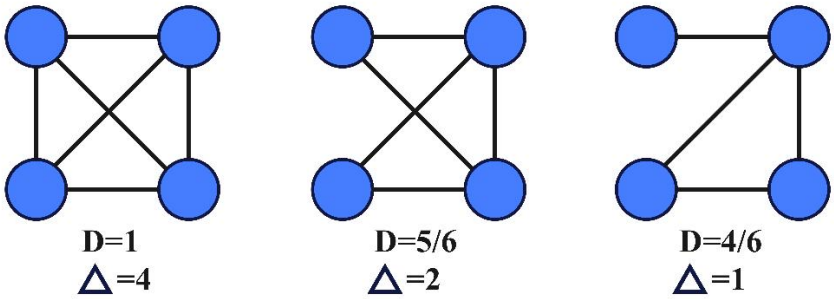

Figure 3 Gradually Remove Network Edges

Figure 3 demonstrates the progressive edge removal process applied to an initially fully connected four-node network. In the middle graph, where one edge has been removed, the network density measures 0.833 with three remaining

triangles. The right graph, with two edges removed, shows a further reduced density of 0.66 and only one preserved triangle. These results clearly indicate a positive correlation between network density and the number of triangular structures. Therefore, the importance of a node can be gauged by the number of triangles it forms with its neighboring nodes.

Therefore, the pseudo code of the algorithm to form a triangle through nodes and their neighbors is as follows:

Algorithm 1: Triangle Calculation

Input: = (,), Set: TempList1[], TempList2[], TempList3[], TempList4[]
Output: Triangle List Final [i, number of Triangle]
1 for ∈ do
2 number of Triangle=0;
3 for ∈ *h* ( ) do
4 for k ∈ *h* (j) do
5 TempList1= getNeighbors(i);
6 TempList2= getNeighbors(j);
7 TempList3= getNeighbors(k);
8 TempList4 = intersection(TempList1, TempList2, TempList3);
9 number of Triangle += Size of TempList4;
10 Update Triangle List Final [, number of Triangle⁄2];
11 end;

3.3 Voting Score Calculation

According to the number of third-order neighborhood triangles of nodes calculated in the upper part, the semi global triangular centrality of nodes will continue to be calculated as follows. According to the characteristic that the connection between a node and its third-order neighbor node is a strong connection in the third-order partition theory, when calculating the triangle centrality, consider the influence of the third-order neighbor node according to different weights from large to small. The specific calculation is as follows：

$$st_c(m) = \sum_{n \in N_m} \triangle_n / 2^d \tag{12}$$

Where, $\triangle_n$ is the number of triangles generated from node n , and N_m is the neighborhood set including node m . The node set consists of node m , the nearest neighbor of node m , and the next nearest neighbor. d represents the distance between node m and neighborhood set N_m . Here, $d=1$ represents the nearest neighbor node of node m, $d=2$ represents the second order neighbor node of node m, and $d=3$ represents the third order neighbor node of node m. According to the characteristics of strong connection between nodes and their third-order neighbors in the third-order partition theory, consider the third-order neighbors of nodes, because the third-order neighbors of nodes are the best choice to balance the spread spectrum cost and performance. If more neighbor steps are considered, the ranking effect will not be significantly improved, or even decline. The specific pseudo code is as follows:

Algorithm 2: Local Semi-Global Triangular Centrality

Input: = (,), Triangle List Final [i, number of Triangle]
Output: Rank [m, LSTC(m)]
1 for ∈ do
2 rank=0;
3 rank= Triangle List Final [ ]/2^0;
4 for ∈ *h* ( ) do
5 rank+= Triangle List Final [ ]/2^1;
6 for ∈ *h* ( ) do
7 rank+= Triangle List Final [ ]/2^2;
8 Update Rank [m, rank];
9 end;

4 EXPERIMENT AND ANALYSIS

4.1 Datasets and Comparison Algorithms

In order to verify the authenticity and effectiveness of the proposed algorithm LSTC, the following performance comparison tests will be carried out. The eight algorithms of the same type are: ECRM、EnRenew、MCDE、Degree、H-index、ISK、K-shell、VoteRank. These algorithms are mentioned above. The six real network data sets required for the experiment are shown in Table 1:

Table 1 Datasets

NetWork	N	M	<k>
Hamster	2426	16631	13.71
NetSci	379	914	4.82
PGP	10680	24316	4.55
Power	4941	6594	2.66
USAir2010	1574	17215	21.87
Yeast	2224	6609	5.94

Where, N is the total number of nodes in the complex network, M is the total number of sides in the complex network, and $< k >$ is the average degree of the network. The relationship between the three network topological properties is $N \cdot < k >= 2 \cdot M$.

4.2 Experimental Model

To accurately evaluate the quality of nodes selected by key node identification algorithms, researchers have developed propagation models. To date, the most widely used models include the Independent Cascade Model (IC), the Infectious Disease Model (SIR), and the Linear Threshold Model (LT), each with its own evaluation metrics. In order to comprehensively and accurately measure the performance of the algorithm proposed in this paper, the Infectious Disease Model and the Linear Threshold Model are selected here. The indicators associated with these models will be described in detail in the following sections.

4.2.1 SIR

The infectious disease model, commonly known as the SIR model [12], is a classic epidemiological framework. Originally developed to quantify the scope and efficiency of virus transmission within a population, it has now been widely used to evaluate the quality of influential nodes identified by impact maximization algorithms [13]. In this case, the set of nodes that initiate earlier and faster propagation is considered to have higher quality. This model divides nodes into three different states: susceptible (S), infected (I), and recovered (R). Lymph nodes in S state are healthy and prone to infection; State I indicates that the node has been infected and is contagious; State R indicates that the node has recovered and gained immunity. It is crucial that each node can only exist in one state at any given time. In addition, state transitions are usually irreversible within a single propagation cycle - specifically, the recovered node cannot become sensitive again, which means that a person is only infected with the virus once.

In the initial stage of simulation, all nodes in the network are set to a sensitive (S) state. Then, the key nodes identified by the algorithm are designated as the initial infection source and transformed into the infection (I) state. Subsequently, each infected node attempts to infect its susceptible neighbors with an infection probability μ , while each affected node can recover with a recovery probability ξ and enter a recovery (R) state. Once restored, immune lymph nodes can no longer be reinfected by any infected neighbors. The values of infection probability and recovery probability are crucial as they directly determine the final scale of transmission. A too high A can cause infection to saturate the network too quickly, thereby masking the relative influence of the initial seed nodes. On the contrary, a too small A may suppress the propagation process and even prevent high impact nodes from triggering meaningful cascades, making it difficult to distinguish their effects.

Based on the structural characteristics of the infectious disease model, two indicators to quantify the influence of nodes are proposed, namely, the infection scale $F(t)$ [14] and the final infection scale $F(t_c)$. Among them, the infection scale refers to the limit value of infected nodes in the network over time. The calculation of the infection scale is as follows:

$$F(t) = \frac{n_{I(t)} + n_{R(t)}}{N} \tag{13}$$

Where, N is the total number of nodes in the network, t is the time, $n_{I(t)}$ is the number of nodes in the network in state I at time t , $n_{R(t)}$ is the number of nodes in the network in state R at time t , where, $N = n_{S(t)} + n_{I(t)} + n_{R(t)}$ 。 It can be seen that the infection scale in the epidemic model represents the proportion of infected nodes and cured nodes infected with viruses in the total number of nodes.

The final infection scale of another quantitative indicator is defined as follows:

$$F(t_c) = \frac{n_{R(t_c)}}{N} \tag{14}$$

Where $n_{R(t_c)}$ is the limit value of the number of nodes in the network with R status over time. It can be seen that the final infection scale in the epidemic model represents the proportion of all infected nodes in the total number of nodes at the end of the virus transmission.

4.2.2 LT

Linear threshold model, also known as LT model [15], is another commonly used influence model. The model has two node states: active state and inactive state. A node in the active state has an activation probability θ , and a node in the inactive state has an activation threshold upper bound θ_{active} . At the initial stage, all nodes in the network are in the inactive state, and the node state selected by the influence maximization algorithm is in the active state. The active node

attempts to activate its direct neighbor with the activation probability θ . The inactive node accumulates the activation probability θ brought by the direct neighbor. If the cumulative value exceeds the activation threshold upper bound, it is activated, and the node state is converted from the inactive state to the active state. The mathematical definition of state transformation is as follows:

$$\theta_{active} \geq \sum_{i \in \Gamma_v} \theta_i \tag{15}$$

4.3 Experimental Result

Before the experimental comparison based on the infectious disease model, the experimental parameters should be tested first. In order to accurately describe the relationship between infection probability and recovery probability in the infectious disease model, the infection rate $\beta = \mu / \xi$ is defined. The infection rate β is proportional to the infection probability ξ and inversely proportional to the recovery probability B. The infection rate will directly affect the scale of virus infection in the infectious disease model. When the infection rate is too high, the scale of virus infection in complex networks is too fast, which is not conducive to the analysis of node influence; When the infection rate is too small, the virus infection scale of the complex network is too slow, and the convergence time of the infection scale is too long, which is not conducive to observing changes.

For this reason, we first designed a comparative experiment between infection rate β and final infection scale $F(t_c)$ to find the infection rate most suitable for the spread of infection scale. The specific experiment is shown in Figure 4. It can be seen from Figure 4 that in Hamster, PGP and USAir2010, no matter what the infection rate is, the final infection scale of LSTC always reaches the maximum. In NetSci, when the infection rate is 0.9, the performance of LSTC is slightly lower than EnRenew. When the infection rate is 0.3, 0.6, 1.2, 1.5, 1.8, the performance of LSTC is higher than EnRenew. In Power, LSTC algorithm is only slightly lower than EnRenew when the infection rate is 0.3. In Yeast, when the infection rate is 0.3 and 0.9, the effect of LSTC algorithm is better than ISK, and in other cases, it is worse than ISK. Even so, LSTC still exceeds the other seven algorithms. It can be seen that when the infection rate is 0.9, the comprehensive performance is the best. When conducting the infection scale experiment, the infection rate is set to 0.9.

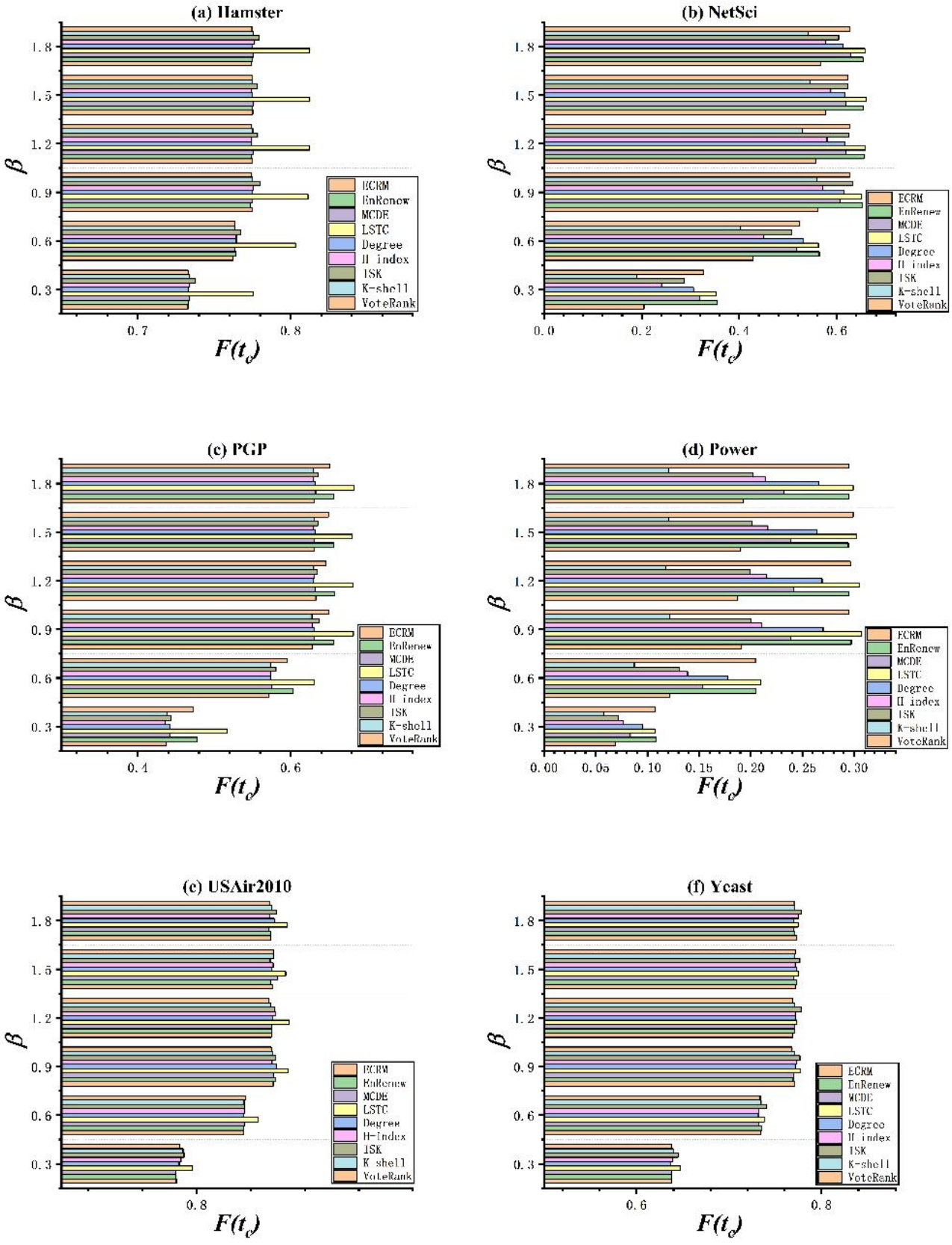

Figure 4 Experiment on Final Infection Scale and Infection Rate

(1) The comparative experiment of infection scale and time based on infectious disease model is shown in Figure 5:

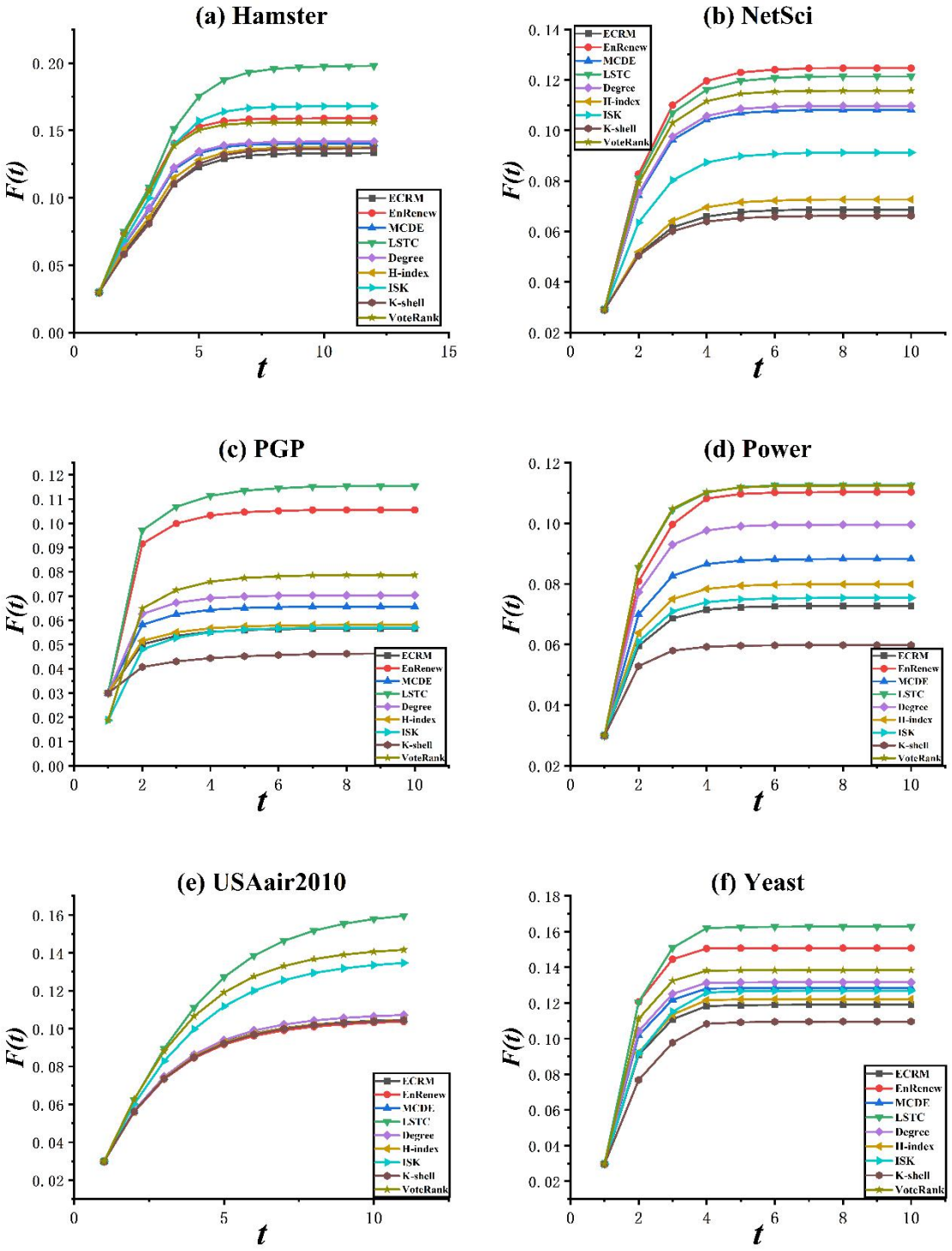

Figure 5 Experiment on Infection Scale and Time

It can be seen from Figure 5 that in the dataset NetSci, as time goes by, the infection scale of key nodes selected by LSTC algorithm is higher than that of ECRM, MCDE, Degree, H-index, ISK, K-shell and VoteRank algorithms. But it is slightly lower than EnRenew algorithm. Thus, the experimental performance of LSTC algorithm is the best.

(2) The comparison experiment between the number of active nodes based on linear threshold and the initial infection ratio is shown in Figure 6:

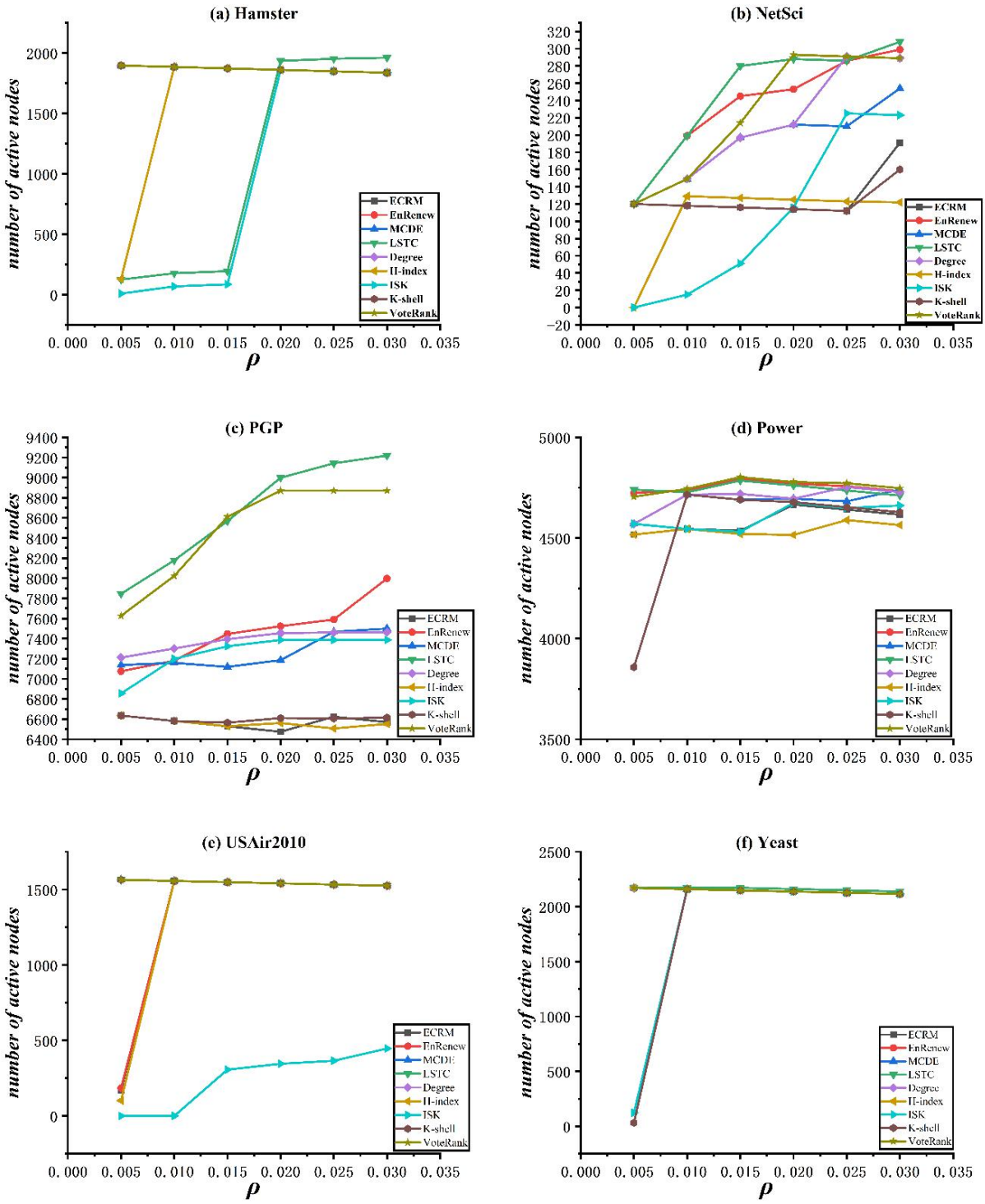

Figure 6 Experiment on the Number of Active Nodes and the Initial Infection Ratio

Where, ρ is the initial infection ratio, which is the proportion of nodes selected by the key node identification algorithm in the total network nodes. It can be seen from Figure 6 that in Hamster, when the initial infection ratio is greater than 0.02, the performance of LSTC algorithm is better than other algorithms, and when it is less than 0.02, it is only better than ISK algorithm. In NetSci, only when the initial infection ratio is 0.02 and 0.025, the performance of the algorithm is slightly lower than that of the VoteRank algorithm. In other cases, it is better than the other eight algorithms. In the dataset PGP, the performance of LSTC algorithm is basically higher than that of other algorithms, reaching the best. In Power, although the performance of this algorithm cannot reach the highest level, the overall trend is better than other algorithms. In the USAir2010 and Yeast data sets, LSTC performance is higher than or equal to other algorithms. In conclusion, LSTC algorithm has the best performance.

5 CONCLUSION

In order to improve the accuracy of key node identification in complex networks, this paper proposes the LSTC algorithm based on the three degree segmentation theory and triangular pattern calculation. This algorithm first introduces the real world stable triangular pattern into the complex network, and measures its importance by calculating the number of triangles formed by nodes and their neighbors. Secondly, referring to the characteristic of strong connection between nodes and their third-order neighbors in the three-dimensional partition theory, the number of triangles formed by nodes and nodes in their third-order local neighborhood is taken as the final calculation basis. By comparing eight algorithms of the same type on six real data sets: experiments based on infection scale and time of infectious disease model, experiments based on the number of active nodes and initial infection ratio of linear threshold model, LSTC algorithm has the best comprehensive performance.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This paper is supported by the Tarim University Euphrates Poplar Talent Introduction Scientific Research Foundation Project (TDZKSS202546).

REFERENCES

[1] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network//Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. Washington, USA, 2023: 137-146.
[2] Wu P, Pan L. Scalable influence blocking maximization in social networks under competitive independent cascade models. Computer Networks, 2017, 123: 38-50.
[3] Leskovec J, Adamic L A, Huberman B A. The dynamics of viral marketing. ACM Transactions on the Web (TWEB), 2007, 1(1): 5-es.
[4] Freeman L C. Centrality in social networks: Conceptual clarification. Social network: critical concepts in sociology. Londres: Routledge, 2002, 1: 238-263.
[5] Lü L, Zhou T, Zhang Q M, et al. The H-index of a network node and its relation to degree and coreness. Nature communications, 2016, 7(1): 10168.
[6] Kitsak M, Gallos L K, Havlin S, et al. Identification of influential spreaders in complex networks. Nature physics, 2010, 6(11): 888-893.
[7] Wang M, Li W, Guo Y, et al. Identifying influential spreaders in complex networks based on improved k-shell method. Physica A: Statistical Mechanics and its Applications, 2020, 554: 124229.
[8] Sheikhahmadi A, Nematbakhsh M A. Identification of multi-spreader users in social networks for viral marketing. Journal of Information Science, 2017, 43(3): 412-423.
[9] Zareie A, Sheikhahmadi A, Jalili M, et al. Finding influential nodes in social networks based on neighborhood correlation coefficient. Knowledge-based systems, 2020, 194: 105580.
[10] Zhang J X, Chen D B, Dong Q, et al. Identifying a set of influential spreaders in complex networks. Scientific reports, 2016, 6(1): 27823.
[11] Guo C, Yang L, Chen X, et al. Influential nodes identification in complex networks via information entropy. Entropy, 2020, 22(2): 242.
[12] Kermack W O, McKendrick A G. A contribution to the mathematical theory of epidemics. Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character, 1927, 115(772): 700-721.
[13] Buscarino A, Fortuna L, Frasca M, et al. Disease spreading in populations of moving agents. Europhysics Letters, 2008, 82(3): 38002.
[14] Ma L, Ma C, Zhang H F, et al. Identifying influential spreaders in complex networks based on gravity formula. Physica A: Statistical Mechanics and its Applications, 2016, 451: 205-212.
[15] Granovetter M. Threshold models of collective behavior. American journal of sociology, 1978, 83(6): 1420-1443.