Swarm and Evolutionary Computation 72 (2022) 101102

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Graph embedding based ant colony optimization for negative influence propagation suppression under cost constraints

Bo-Lun Chen [a,b], Wen-Xin Jiang [a,*], Yong-Tao Yu [a], Lei Zhou [a], Claudio J. Tessone [b]

[a] College of Computer Engineering, Huaiyin Institute of Technology, Huaian 223300, China
[b] Blockchain and Distributed Ledger Technologies, Institute of Informatics, University of Zürich, Zürich CH-8050, Switzerland

ARTICLE INFO

Keywords:
Social networks
Cost constraint
Influence suppression
Ant colony optimization
Graph embedding

ABSTRACT

In the information spreading mechanism of social networks, the influence propagation of information sources often has different effects on different users. How to effectively suppress the negative effects is particularly important. In the case of unknown network propagation principle, this paper introduces the idea of swarm intelligence, which utilizes the positive feedback mechanism of ant colony to simulate the propagation of negative influence, and finds a set of high-value and low-cost suppression nodes. On this basis, the graph embedding technique is used to obtain the new relationships between nodes in the network, and the new relationships between the nodes are used as heuristic information for the ant colony algorithm. Experiments show that our algorithm can not only find the set of inhibitory nodes with limited cost, but also effectively limit the spread of negative influence in the network compared with other algorithms. The research of this paper can not only enrich the theoretical research results of influence maximization, but also play an important role in the analysis of network topology, as well as in the fields of epidemic prevention and control, rumor propagation and so on.

1. Introduction

In recent years, with the vigorous development of online social networks, thousands of people produce unprecedented massive data through various forms of medias, and the data present the trend of multi-source and isomerization. At the same time, the popularity of online social networks has aroused people's great interest in information spreading. A piece of information may be rapidly popularized through ”word-of-mouth” among friends in the network. In the information spreading mechanism of social networks, information spreading among different users is often affected by the influence of users [1]. Therefore, it is very necessary to carry out social network analysis research, which has attracted extensive attention from computer science, physics, epidemiology, and other fields [2,3].

Influence maximization (IM) is an important research topic in social network analysis, which aims to find the key subset of users to maximize influence propagation under a specific propagation model [4,5]. Its task is to select k users (also known as seed set) to spread information through the contact between friends in social networks, and these users can affect a large part of users of social networks as much as possible. For example, in the case of viral marketing, the company hopes to increase the sales of its products by relying on ”word-of-mouth” recommendation and maximize the publicity effect by tapping the influence of important customers in the network [6]. Influence blocking maximization (IBM) is an extension and expansion of the traditional influence maximization problem. Different from the traditional influence maximization, IBM is mainly applied to the case where there is competition between the nodes in the network. In previous studies, most researchers have solved the problems of non-competitive networks. In non-competitive networks, there is no competitive relationship between users, but in the real world, there are a variety of competitive relationships between users, and the competitive spreading of a variety of information is very common in the network [7,8]. In the process of influence propagation, there are positive influences such as support and praise, as well as negative influences such as bad comments and rumors, of which many negative news spread rapidly and wantonly in the network. The COVID-19 outbreak has generated a large number of rumors on the Internet and has been widely spread among users. For example, ”Yansong Bai, host of CCTV News Channel, will host a special program on COVID-19 at 9:30 p.m., academician Nanshan Zhong will be invited to introduce the situation related to the epidemic.”. For a time, WeChat groups, circles of friends, forums and even some big V bloggers were forwarding the news. Later, it was confirmed that the news was a rumor. It can be seen that negative influence has the characteristics of short propagation time, fast propagation speed, and wide range of influence.

Therefore, it is especially important to take some measures to stop the spread of negative information such as rumors and to trace the source of rumors. This is of great theoretical importance for analyzing,

* Corresponding author.
E-mail address: 212006560474@hyit.edu.cn (W.-X. Jiang).

https://doi.org/10.1016/j.swevo.2022.101102
Received 26 November 2021; Received in revised form 7 April 2022; Accepted 26 May 2022
Available online 31 May 2022
2210-6502/

understanding and predicting the topology, function and dynamic behavior of complex networks, providing theoretical support for virus propagation, public opinion control and disinformation control in government departments, and providing strong theoretical support and practical guarantee for social security, stability and economic development [9,10]. To summarize, our work makes the following contributions:

(1) The idea of swarm intelligence is introduced, and the positive feedback mechanism of ant colony is used to simulate the spread of negative influence.
(2) Graph embedding technology is used to reconstruct the relationship between nodes in the network.
(3) This method can select cost-effective suppression nodes with limited cost, and can effectively prevent the spread of the node's negative influence in different communities.
(4) This research can play an important role in the fields of epidemic prevention and control, rumor spreading and so on.

2. Related works

2.1. Traditional methods

Information spreading models are the basis for the study of influence maximization. Domingos and Richardson [11] first studied the influence propagation on social networks and analyzed it using data mining related technologies. Kempe et al. [12] formulated the influence maximization problem based on independent cascade model (IC) and linear threshold model (LT), and proposed a greedy algorithm similar to the optimal result $(1 - 1/e)$. In recent years, these two probability models have become important models to study influence propagation. Many algorithms designed for the influence maximization problem and their variants are based on IC or LT. In the influence propagation process of social networks, the negative influence propagation suppression maximization problem is actually to prevent the influence propagation of its competitive entities by selecting some seed nodes. He et al. [13] called this problem the influence blocking maximization problem. Also, they proved that the objective function of IBM is submodular in the competitive linear threshold model and that the greedy algorithm can achieve the optimal solution.

However, the time consumption of greedy algorithm will increase sharply with the increase of the network scale. Therefore, Leskovec et al. [14] proposed the CELT (cost effective lazy forward) algorithm which takes into account the decreasing edge payoffs, and they excluded the nodes with small edge payoffs in the previous round when calculating the edge payoffs in the next round, effectively reducing the calculation time of the greedy algorithm. Tong et al. [15] found that the algorithm combining the classical greedy algorithm and Monte Carlo simulation ran too long, and proposed a random approximation algorithm to improve the operation efficiency while ensuring the performance. Because the greedy algorithm is slow and not scalable, Wu and Pan [16] designed a heuristic algorithm according to the maximum influence tree structure, which runs much faster than the greedy algorithm although its effect is close to that of the greedy algorithm.

2.2. Methods based on node and edge importance

Another important approach in IBM's study of the problem is by blocking nodes and links that have played an important role in the spread of negative information [17]. The goal of this type of approach is to select and remove k key nodes in the social network so that the spread of negative information can be blocked to the maximum extent possible after removing the seed set. Khalil et al. [18] proposed a greedy edge-deletion-based algorithm to address the problem of rumor blocking, that is, to remove a set of k edges such that rumor spread is minimized under the LT model. Zhang and Prakash [19] proposed three effective polynomial-time heuristic algorithms, DAVA, DAVA-prune and DAVA-fast, which can help public health experts to make real-time scenarios based on the current epidemic distribution. Arazkhani et al. [20] proposed the Centrality_IBM algorithm, which uses three different centrality strategies, including closeness centrality, betweenness centrality and degree centrality to find k positive nodes to prevent the propagation of negative influence. Lee et al. [21] proposed the influence distribution redirection algorithm by analyzing the potential influence trend of nodes in the process of influence diffusion. Peng et al. [22] proposed a new containment model based on an influence maximization algorithm. The model first assesses the influence of nodes by introducing a social relationship graph, then finds the most influential nodes using the election system, and finally takes immunization measures for the top k influential nodes to prevent the spread of negative influence. Kuhlman et al. [23] proposed an edge-covering heuristic algorithm under the discrete dynamical system model. Xue et al. [24] argued that convincing influential people to act as initial spreader is costly and difficult and proposed a risk-aware metric to identify the most effective spreaders in real networks based on assumption that the activation of large-degree nodes carries a higher risk than that of small-degree nodes.

2.3. Methods based on community structure

At present, there is still a part of the work on the network-based community structure. Arazkhani et al. [25] proposed a method to find a good candidate subset of nodes for diffusion of positive information using fuzzy clustering and concentration measure. To improve the efficiency of the IBM algorithm, based on the locality of influence diffusion in social networks, Lv et al. [26] proposed a community structure-based IBM algorithm CB_IBM. On the basis of the Ising model, Wang et al. [27] proposed a dynamic negative influence diffusion model, which incorporates the global negative influence prevalence and individual tendencies. Zhu et al. [28] found that location information can play an important role in influence propagation and proposed two heuristic algorithms LIBM-H and LIBM-C based on quadtree index and maximum tree structure. Considering that bridge ends often have important values in the community structure, Fan et al. [29] built a reverse search tree to find all nodes affecting bridge ends and found the minimum number of key node sets by transforming it into a minimum set coverage problem. Gong et al. [30] proposed a local search strategy based on similar nodes in the community, which can effectively accelerate the convergence of the algorithm. Li et al. [31] proposed an influence index to measure the community diversity, and then proposed greedy and previous algorithms to calculate the influence maximization. Experiments show that this method can also achieve good results in networks with unclear community structure. In order to solve the challenge of finding important nodes in a large social network, Sivaganesan [32] proposed an interest-based parallel social behavior algorithm. This algorithm integrates user behaviour and interests and enables parallel computation through community structures, improving the computational efficiency of real-world large-scale social networks.

From the current research status at home and abroad, it can be seen that in the research of negative influence suppression, traditional methods treat the characteristics of positive and negative influence propagation equally. However, in the real environment, negative influences spread faster and more widely than the spread of positive influences. In addition, some of these existing methods ignore the cost of deleting important nodes, and some algorithm parameters need to be manually determined in advance. Therefore, in the process of suppressing the propagation of negative influence, how to measure the propagation speed and scope of the negative influence and the comprehensive evaluation of the suppressed nodes is particularly important. This paper combines the idea of graph embedding in deep learning to map and reconstruct the network from high-dimensional to low-dimensional, and uses a group of optimization algorithms with positive information feedback and heuristic search characteristics to measure the influence of online

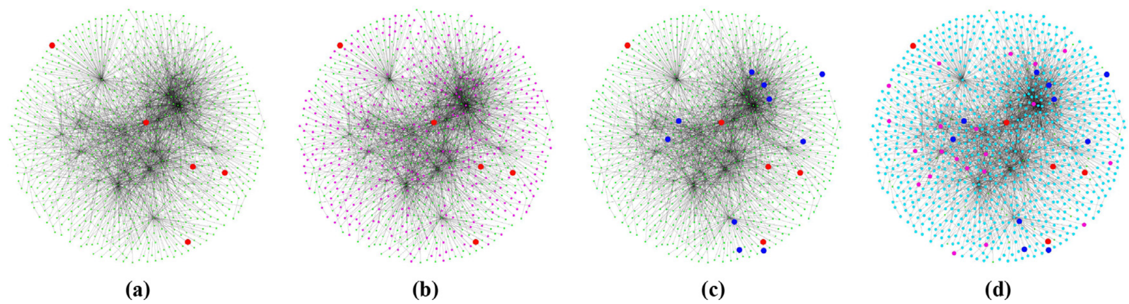

Fig. 1. Schematic diagrams of negative influence propagation and suppression. Among them, the red nodes represent the negative influence nodes in network communication, the green nodes are the healthy nodes not affected, and the pink nodes are the nodes affected by the negative influence. Blue nodes are the selected suppression nodes, and light blue nodes are the nodes that can be covered by the suppression nodes. (a) The initial network state when the negative influence is not propagated; (b) As time changes, the network state after negative influence propagation; (c) The state after adding suppression nodes in the original network; (d) The network state in which nodes prevent the propagation of negative influence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

social networks to find high value, low-cost key node set, design an effective method to suppress the spread of negative influence.

3. Problem modeling

3.1. Description of the problem

Online social network can be regarded as a mapping of people's real social life on the Internet. In this paper, the influence propagation of social network is simulated with the structure of graph. Each user under social relationship is regarded as a node in the network graph, and the relationship between different users is connected with edges. In this way, the social network can be defined as a graph composed of nodes and edges. The influence propagation diagram of different types of nodes in the network is shown in Fig. 1.

As can be seen from Fig. 1, when the original network does not suppress the negative influence, most of the healthy nodes in the network will be infected by the negative influence. However, after selecting some suppression nodes, it can block the spread of negative influence in the network and ensure the healthy development of the network.

In terms of mathematical definition, the network can be defined by $G = (V, E)$, where V represents the set of all nodes in the network and E represents the set of edges between these nodes. From the perspective of information spreading, the nodes in the graph generally have two states: inactive state and active state. Only the nodes in the active state have the ability to affect other nodes. A node in the inactive state can only be activated by the neighbor node in the active state if it wants to be converted to the active node. If there are two nodes u and v in social network G, and u and v are not the same node, $(u, v) \in E$ indicates that there is a connecting edge between u and v, and the influence can be directly transmitted between them, and the propagation probability is expressed by p. Where, n represents the total number of nodes in network G, i.e. $n = |V|$, and m represents the total number of edges in network G, i.e. $m = |E|$. Then, the influence maximization problem is transformed into finding k seed nodes in G, so that they can prevent the negative influence of nodes from spreading in the network.

In addition, if we want to select some nodes to suppress the spread of the negative influence of the nodes in the network G, we need to pay a corresponding price, because the cost of each node is different. For example, a company wants to expand its influence through celebrity endorsements, but since each celebrity's appearance fee is different and the company's advertising expenses are limited. Therefore, a reasonable strategy needs to be devised to expand the influence of the company. In this paper, we set $cost(v)$ to be the cost of node v and the total cost of cost to be Q. Let $I(S_N, S_P)$ be the expected value of the number of vertices activated by the negative seed set when S_P is the negative seed set and S_N is the positive seed set. Therefore, the purpose of the negative influence suppression problem under the cost constraint is to select an optimal positive seed set S^* in the vertex set $V \backslash S_N$, so that $I(S_N, S^*)$ is the smallest.

$$S^* = \arg \min_{v \in (V \backslash S_N)} I(S_N, S_P), \sum_{v} cost(v) \le Q \tag{1}$$

3.2. Influence diffusion model

Kempe et al. [12] proposed that the famous independent cascade model can be regarded as a probability model with better universality and flexibility. Therefore, this paper uses it as the communication model of negative influence communication inhibition in the process of influence communication. In the independent cascade model, the node has and only has two states, active state and inactive state. The active node represents the user who has been affected by negative information and tries to spread to other nodes. An inactive node represents a user who has not received a negative message or has received a negative message but ignored it.

The basic assumption of the model is that whether the behavior of node u trying to activate its adjacent node v is successful or not is an event with a probability of $p(u, v)$. And the probability that a node in an inactive state is activated by a neighbor node that has just entered the active state is independent of the activities of other neighbors who have tried to activate the node before. In addition, any node u in the network has only one chance to try to activate its neighbor node v. No matter whether it is successful or not, although u itself remains active in the future, it no longer has influence. This kind of node is called non influential active node. The propagation process of information in the independent cascade model can be defined as the following three steps:

(1) At the initial stage of information spreading, that is, at time t_0, a few active nodes will be set in the network as the initial active node set S_N. All nodes in S_N are active and all nodes except S_N are inactive.
(2) In the next time $t > 0$, the node activated at time $t-1$ will attempt to activate the inactive neighbor node with probability $p(u, v)$, and the attempt of node u to activate the inactive neighbor node v is independent of the attempts of other active nodes v. If node v is successfully activated, node v will be converted to the active state at time t.
(3) When no node in the graph changes state at a certain time, the propagation process of the independent cascade model ends.

4. Methodology

4.1. Basic idea of algorithm

In the scheme design of negative influence propagation suppression, the key is how to simulate the propagation path of negative influence

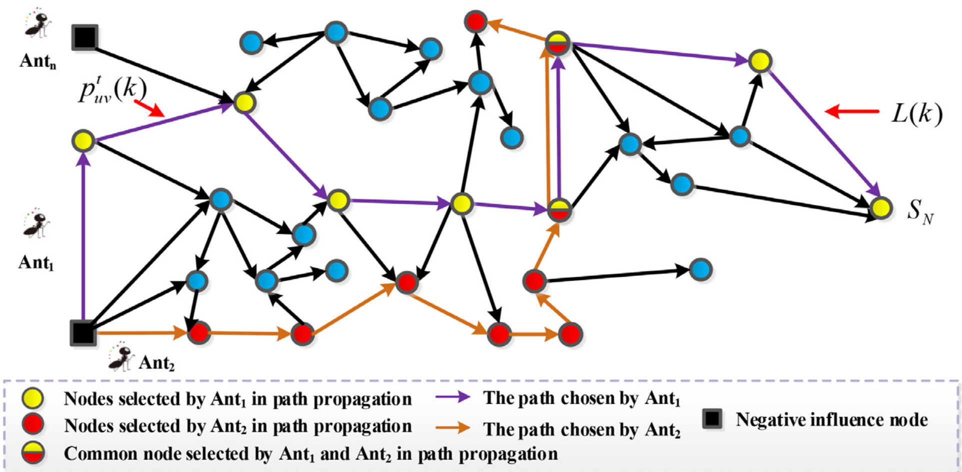

Fig. 2. Schematic diagram of the propagation path of ants.

Table 1
The framework of Ant Colony Optimization algorithm based on Graph embedding.

Algorithm 1 ACO based on graph embedding(ACO-GE)

Input: Adjacency matrix A
  Maximum number of iterations NC
  Number of ants m
  Length of ant selection path L
  Negative influence node set S_N
Output: Suppress node set S_P
Begin
1. Parameter initialization: $\tau \leftarrow A, \eta \leftarrow Deepwalk(A)$;
2. **for** i=1 to Nc do
3. **for** k=1 to m do
4.   Path selection: $L_{v\in S_N}(k) \leftarrow p^t_{uv}(k), p^{t+1}_{uv}(k), \ldots, p^{t+L}_{uv}(k)$;
5.   Evaluation path: $\Delta\tau^t_{uv}(k) \leftarrow f(L_{v\in S_N}(k))$;
6.   Pheromone update:$\tau^t_{uv} \leftarrow \Delta\tau^t_{uv}(k)$;
7. **end for**
8.**end for**
9. Node candidate set:$S_C \leftarrow v \in L_1(k), L_2(k), \ldots, L_N(k)$;
10.Suppress node set: $S_P \leftarrow rank(v) \leftarrow cost(v), v \in S_C$

nodes and mine the information of key nodes in the network, and finally complete the selection of suppression nodes based on this. This paper mainly introduces the idea of swarm intelligence and graph embedding, and uses the positive feedback mechanism in ant colony algorithm to select the suppression nodes.

The main steps of the algorithm are as follows: firstly, the hidden information features of the network are learned by using Deepwalk graph embedding technology to obtain the potential features of the nodes. On this basis, the similarity between different nodes is measured and used as a heuristic factor in the ant colony algorithm. Next, ants will simulate and iterate the influence propagation path from the negative influence node to form a key propagation path set. Finally, the important nodes in the critical paths are obtained, and the node costs are combined to make the selection of suppression nodes. The algorithm framework is shown in Table 1.

4.2. Detailed steps of the algorithm

4.2.1. Calculation of critical path of ant colony algorithm

In this step, we set the number of ants as m and the set of negative influence nodes as S_N. At time t, we randomly scatter ants on the negative influence nodes, and the ants choose the path according to the probability $P^t_{uv}(k)$. After time t, the ant completes the traversal of the whole network from the negative seed node S_i and forms a path, which is represented by $L_i(k)$. The schematic diagram of ant selection path is shown in Fig. 2.

4.2.2. Calculation of critical path of ant colony algorithm

We set τ^t_{uv} as the pheromone between nodes u and v at time t, and η^t_{uv} as the heuristic factor between nodes to reflect the expected degree of ants from node u to node v. In the initial stage, the pheromone on each edge can be positively correlated with the weight on the edge. We introduce graph embedding technology to measure the importance of the edge. The specific steps will be introduced later. Next, the ant selects the path according to the following probability formula:

$$p^t_{uv}(k) = \begin{cases} \frac{(\tau^t_{uv})^\alpha \cdot (\eta^t_{uv})^\beta}{\sum_{l\in allowed_k} (\tau^t_{ul})^\alpha \cdot (\eta^t_{ul})^\beta} & v \in allowed_k \\ 0 & v \notin allowed_k \end{cases} \tag{2}$$

Where $allowed_k$ is the node set that the kth ant can select, and α and β are the parameters that affect the node selection by pheromone and heuristic information on the regulation path, respectively. We can see that when $\beta = 0$, it shows that the path selection process of ants is only related to pheromones, that is, it meets the rules of the traditional influence propagation model. If $\alpha = 0$, the algorithm becomes a pure greedy algorithm, so we need to reasonably optimize. Next, we need to update the pheromone on each path. The update formula is as follows:

$$\tau^{t+1}_{uv} = (1-\rho)\tau^t_{uv} + \Delta\tau^t_{uv} \tag{3}$$

After an ant travels a path, the original pheromone on the edge will gradually evaporate over time, but each ant will leave additional pheromones after choosing this path. Therefore, we use $\rho \in (0, 1]$ to represent the evaporation coefficient of the pheromone on each edge, $\Delta\tau^t_{uv}$ represents the increment of the pheromone between paths u and v, and we use the following publicity:

$$\Delta\tau^t_{uv} = \sum_{k=1}^{m} \Delta\tau^t_{uv}(k) \tag{4}$$

Where $\Delta\tau^t_{uv}$ is the pheromone increment released by ant k between edges u and v. The value of $\Delta\tau^t_{uv}(k)$ is defined differently in different ant colony models. In this paper, we define it as follows:

$$\Delta\tau^t_{uv}(k) = \sum_{v\in L(k)} d_1(v)^i \cdot d_2(v)^j \tag{5}$$

Where $d_1(v)$ is the degree of node v, $d_2(v)$ is the number of all second-order neighbors of node v, and i and j are the coefficients between them. M ants form a path in the process of each iteration. After this process, they return to the starting point and continue to repeat the process. After several iterations, the ant starting from each node will eventually form a path. We call it the critical path of the negative influence node, and then it will eventually form $|S_N|$ critical paths.

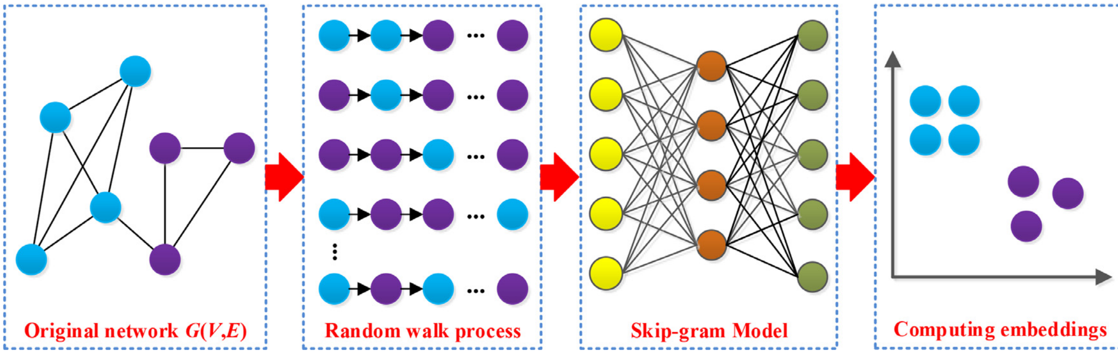

Fig. 3. Illustration of Deepwalk method. Deepwalk first samples the node sequence in the graph by random walks on the original network G, and then learns the embedding through the Skip-gram model. The leftmost is the original input network graph, and the rightmost is the output result. It can be seen that the relationship between the close nodes in the original graph is also well presented in the two-dimensional plane after embedding by DeepWalk.

4.2.3. Calculation of heuristic factors

In the process of calculating the heuristic factor, firstly, we use the Deepwalk based graph embedding method and take the original network as the input to obtain the vector representation of the vertices in the network [33]. Finally, the similarity between nodes is calculated as the heuristic factor of the ant colony algorithm. The graph embedding method of Deepwalk mainly learns the social representation of a network by truncated random walk. Its purpose is to map the network model to a low dimensional vector space, so that it can retain the structural information and potential characteristics of the graph model, and more hidden features can play an important role in the future influence communication task. The schematic diagram of the Deepwalk is shown in Fig. 3.

As can be seen from Fig. 3, Deepwalk is mainly composed of two parts: random walk and generating representation vector. In the process of random walk, let a particle start from node u at time t, and define $\pi^t(u,v)$ as the probability that the particle just walks to node v at time $t+1$, then the system evolution equation can be obtained:

$$\pi^t(u,v) = P\pi^t(u), 0 \le t \le l \tag{6}$$

Where $\pi^0(u)$ is a vector of $N \times 1$, only the uth element is 1, and the other elements are 0. $P = [p(u,v)]$, $p(u,v) = a(u,v)/k(u)$. $a(u,v)$ is the element in adjacency matrix A, and $k(u)$ is the degree of node u. Through this step, each node u generates a sequence W_u with a path length of l according to the random walk strategy.

Then, with the help of the idea of natural language processing, the vertex path W_u obtained by random walk is regarded as a sentence composed of words. Here, we use Google's word2vec tool, then, select the Skip-gram model and take the one hot vector of random walk vertices as the input, and finally get the d-Dimension vector of each vertex [34].

Let's illustrate this step with an example. If we make the number of nodes in the network N, then all our nodes are numbered with 1-N. For example, node 3 generates a sequence with a path length of 10 according to the random walk strategy, which is $W_{u_3} = \{3, 10, 4, 5, 7, 9, 1, 25, 33, 13\}$. Then, first, we choose 4 in the sequence as our input word; after having input word, we define a parameter w called window size, which represents the number of words we select from the left or right of the current input word. If w= 2, we can finally get the words in the window as [3,10,4,5,7]. Therefore, we will get four groups of training data in the form of (input word, output word), namely (4,3), (4,10), (4,5), (4,7). The schematic diagram is shown in Fig. 4.

The blue box is the input word and the window size is 2. Finally, we build a neural network based on the training data. Based on these training data, the neural network will output a probability distribution to show how likely each word in the dictionary is to appear at the same time as input word. For example, if the probability of 10 and 4 in the output probability of the final model is large, it indicates that 10 and 4 are highly correlated. Specifically, we can build a vocabulary with size

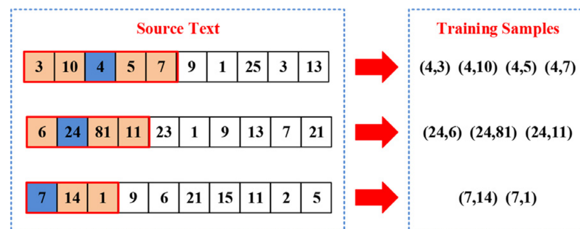

Fig. 4. Example of training set generation.

N based on all nodes. If We assume that node 4 is the input word, it can be represented by a matrix I = [0,0,0,1,0,...,0] of 1-dimensional N columns, where node 4 is set to 1 and the rest is 0. We take it as an input vector. Then, if the input of the model is an N-dimensional vector, the output is also an N-dimensional vector, which contains N probabilities. Each probability represents the probability that the current word is output word in the input sample. The structure of the neural network is shown in Fig. 5.

In this paper, we set 10 features to represent a word, then the weight matrix of the hidden layer should be n rows and 10 columns, that is, the hidden layer has 10 nodes. Therefore, the main purpose of this model is to obtain the weight matrix of hidden layer in the neural network through neural network training.

Finally, we obtain the abscissa and ordinate of all nodes on the plane graph through graph embedding. We calculate the Euclidean distance between any two points and normalize it as the heuristic factor of the ant colony algorithm.

5. Result and discussion

5.1. Data set

We validate the algorithm using six different data sets, including an ER random network generated by computer simulations and five real data sets with community structure: Karate, Soccer, Wiki, Soc, and Email (http://snap.stanford.edu/data/). According to the tightness of the connection between different nodes in the network, the five real data sets regard the network as composed of different communities, in which the connection between nodes in the community is closer, and the links between nodes in different communities are sparse. Among them, Karate data set is a social network describing the friendship between 34 members of American university karate club in the 1970s. The football data set is a social network created for American football matches between American universities during the fall regular season of 2000. In the wiki data set, if a web page node has a link to other web page nodes, it is considered that there is an edge between the two web pages. Soc is who-

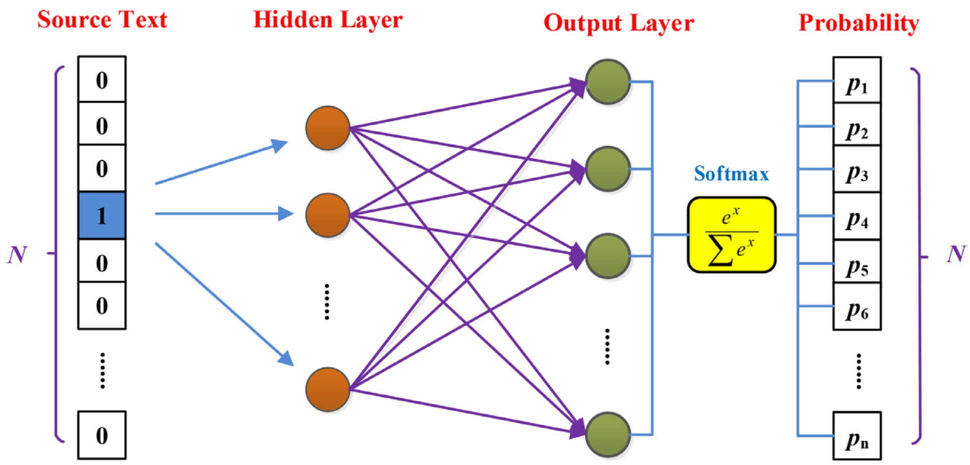

Fig. 5. The architecture of the Skip-gram module.

Table 2
The topological properties of six different data sets.

Data sets	N	E	$D_{\max}$	D_A	CC	DE
Synthetic	81	246	11	5.66	0.175	0.066
Karate	34	78	17	4.59	0.588	0.139
Football	115	613	12	10.66	0.403	0.094
Wiki	2405	8591	250	7.14	0.246	0.003
Soc	3783	14,124	511	7.47	0.277	0.002
Email	1005	16,706	374	33.25	0.473	0.033

trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. Email data set represents the core of the email-EuAll network, which also contains links between members of the institution and people outside of the institution.

Table 2 shows the topological properties of different data sets, where N is the total number of nodes in the data set, E is the total number of edges between nodes in the network, $D_{\max}$ is the number of nodes with the most degrees out of all nodes, and D_A is the average degree of all nodes. CC is the clustering coefficient indicating the degree of clustering of nodes in the network. The higher the aggregation coefficient, the closer the connection between nodes. DE is the network density, indicating the ratio between the actual number of connections and the possible number of connections between nodes. The higher the network density, the denser the connections between nodes.

5.2. Result analysis of graph embedding

In the heuristic factor calculation of the ant colony algorithm, we first perform Deepwalk operation on the original network diagram, and the experimental results are shown in Figs. 6 and 7.

Among them, Fig. 6 is the topology of the original network. In this figure, we use the same color to represent the nodes belonging to the same community, but the hidden relationship between nodes is not completely expressed through the topology. For example, the relationship between different nodes in the same community is only measured by one edge, which does not show the hidden information such as the tightness of nodes. Figure 7 is a vertex vector graph obtained by the graph embedding method of Deepwalk. It can be seen from the graph that the network model is mapped to a low-dimensional vector space, retaining the structural information and potential characteristics of the original graph model. These hidden features will play an important role in the design of the influence communication model.

Next, we calculate the Euclidean distance between any nodes according to the vector, and then set the heuristic factor between nodes through the Euclidean distance. Among them, Deepwalk learns the social representation of a network by truncating random walk, which can also get better results when there are few network annotation vertices. Because the random walk process can be processed in parallel. For a large network, we can start a certain length of random walk at different vertices at the same time, and multiple random walks can be carried out at the same time, which can reduce the sampling time. In addition, the evolution of the network is usually the change of local points and edges, which will only affect some random walk paths. Therefore, it is not necessary to recalculate the random walk of the whole network every time in the process of network evolution. Therefore, the graph embedding method has certain adaptability.

5.3. Analysis of influence propagation suppression results

In order to verify the performance of the proposed algorithm, we compare the proposed algorithm with five other classical algorithms. The five algorithms are: Random algorithm, DegreeCentrality algorithm, PageRank algorithm [35], CI algorithm [4] and K-core algorithm [36]. Among them, the Random algorithm is one of the commonly used comparative experiments for influence suppression, which suppresses the set of seed nodes by randomly selecting k nodes in the network as the set of seed nodes. The DegreeCentrality algorithm sorts all nodes based on their degree, the higher the degree, the higher the influence. The algorithm selects the node with the highest degree among all nodes to join the seed set in each round until the number of nodes reaches k. The PageRank algorithm calculates the PageRank value of each node through the connection relationship of the nodes in the network, and then sorts its importance according to the size of the value. CI algorithm is a scalable algorithm that takes into account collective impact effects. K-core algorithm assigns an integer index or coreness to each node, representing its location according to successive layers in the network.

In Eq. (1), the goal of the algorithm is to select a positive seed set s^* to minimize $I(S_N, S^*)$. In order to reflect the suppression effect of negative information propagation, this paper defines the suppression effect of the algorithm as $N_P = I(S_N, \varnothing) - I(S_N, S_P)$, that is, the suppression effect of the suppression node set is the average negative node activation number of the network without the suppression node set minus the average negative node activation number of the network under the action of the suppression node set. In order to ensure the experimental effect, each algorithm is performed 1000 times, and the average value is obtained and two decimal places are retained as the final expected suppression effect. For each network, we randomly select 5% of the nodes from the network to generate a negative seed set S_N, select the value of the suppression node set S_P according to the size of the network, and compare the suppression effects of the six algorithms. The experimental results are shown in Fig. 8, S_P is the number of suppression nodes and the N_P is the expected suppression effect of the suppression node set.

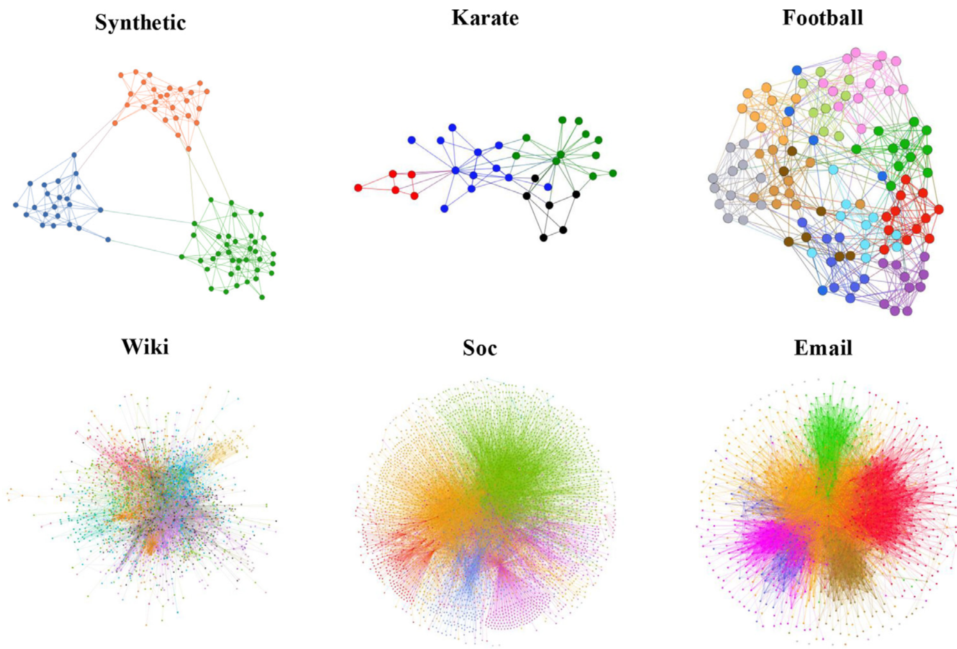

Fig. 6. Visualization diagrams of six different networks.

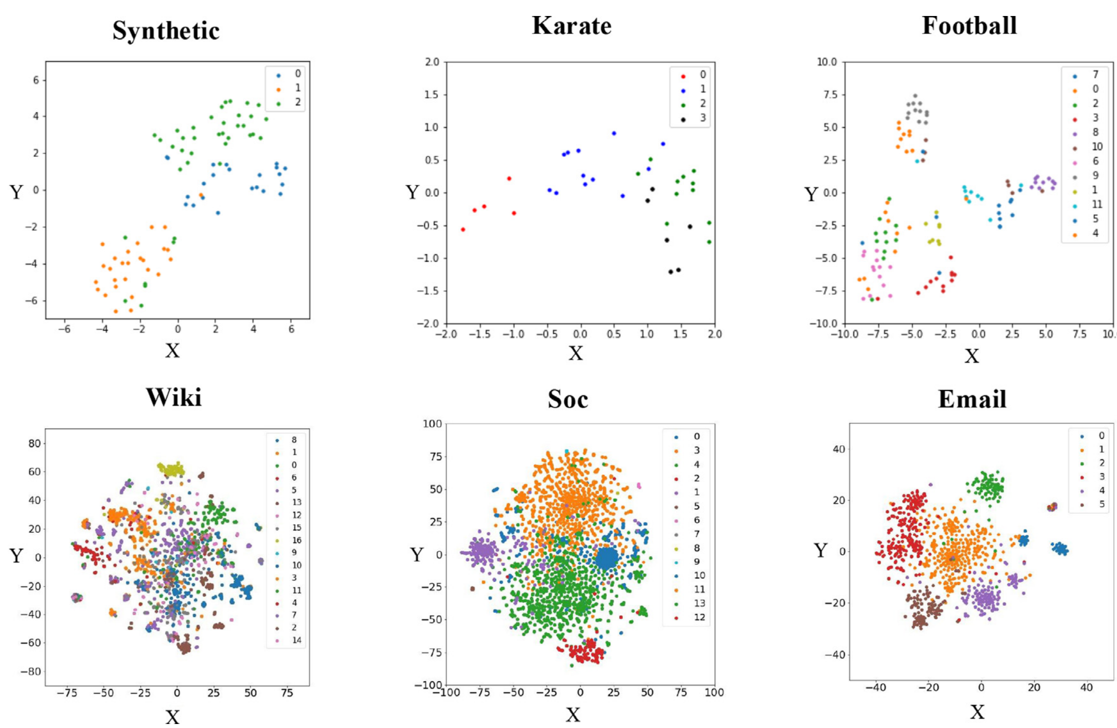

Fig. 7. Two dimensional graph of six different networks after graph embedding operation.

It can be seen from Fig. 8 that in the synthetic data set, the key nodes selected by the algorithm in this paper have better suppression effect in the network. For example, when we set the number of suppression nodes to 5, Random, DegreeCentrality, PageRank, K-core and CI algorithms affect about 52 nodes, 56 nodes, 58 nodes, 42 nodes and 53 nodes, respectively, while the ACO-GE algorithm can affect 63 nodes.

Similarly, in the real data sets, the algorithm in this paper can also achieve better performance. For example, in the Karate data set with a small number of nodes, when the number of suppression nodes is 8, Random, DegreeCentrality, PageRank, K-core and CI algorithms affect about 18 nodes, 20 nodes, 19 nodes, 19 nodes and 20 nodes, respectively, while the ACO-GE algorithm can affect about 22 nodes. In the wiki data set with a large number of nodes, when the number of suppression nodes is 3, Random, DegreeCentrality, PageRank, K-core and CI algorithms affect about 273 nodes, 584 nodes, 627 nodes, 401 nodes and 277 nodes, respectively, while the ACO-GE algorithm proposed in this paper can affect about 656 nodes.

It can be seen that Random algorithm randomly selects nodes without considering the network topology, which has poor effect on the data set with community structure. In addition, the DegreeCentrality algorithm considers the local topology characteristics of the network, and the PageRank algorithm focuses on the edge relationship between nodes. They perform well on small data sets, but behave generally on large data sets. Whether it is a small data set or a large data set, the

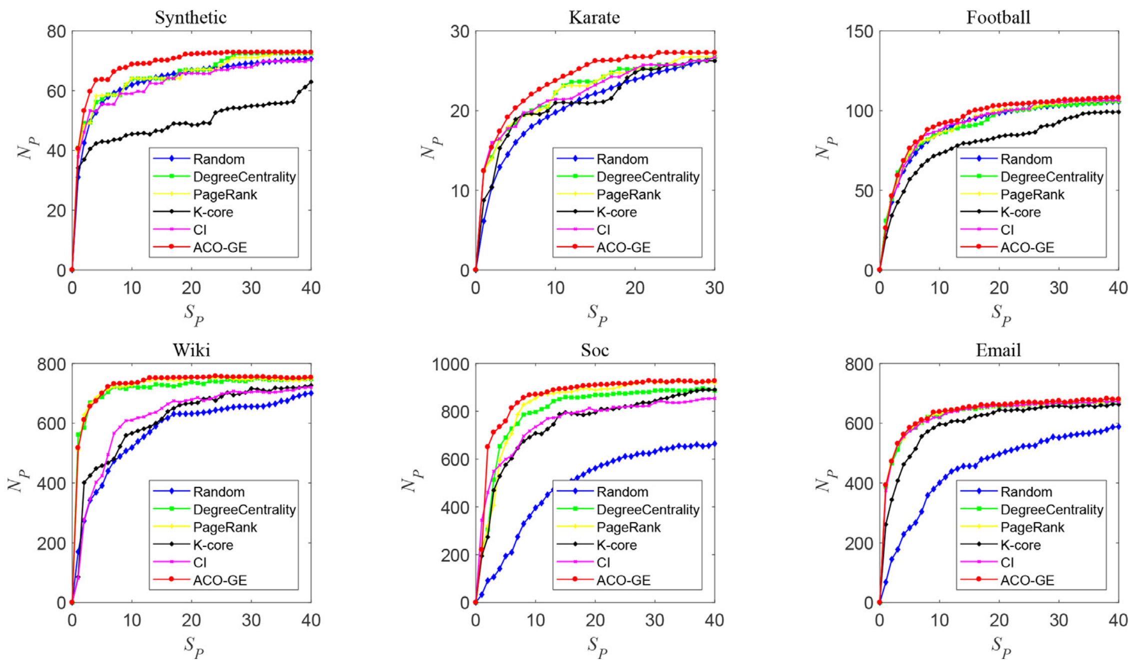

Fig. 8. The change trend of the number of nodes that can be suppressed by different algorithms with the increase of the number of suppressed nodes in different data sets.

influence suppression effect of the algorithm proposed in this paper is relatively good. Therefore, the ACO-GE algorithm has better robustness and scalability.

5.4. Cost performance analysis of suppression nodes

In the process of negative influence propagation, it is necessary to select nodes with high cost performance as the suppression node set, so as to maximize the suppression of influence with limited cost. For example, in real-world social platforms, social accounts with more fans may have greater influence. If such accounts want to spread some information, it needs a high cost. Therefore, we need to consider how to obtain the maximum influence effect with limited cost. According to the previous description of social networks, we define the cost of node v as follows:

$$cost(v) = k(v)/d_{\max} \tag{7}$$

Where $k(v)$ is the degree of node v and $d_{\max}$ is the maximum degree of node among all nodes. It can be seen that the cost value of node v is directly proportional to the degree of the node. The greater the degree of the node, the higher the cost of selecting the node as the node to suppress the node. Similarly, for each network, 5% nodes are randomly selected from the network as the initial negative seed set, and the expected suppression effects of the six algorithms under different costs are calculated. In order to reduce the experimental error, 1000 times are carried out for each algorithm under each cost, and the average value is taken as the final expected suppression effect. Figures 9 and 10 show the cost performance comparison results of suppression nodes of different algorithms in different data sets.

As can be seen from Fig. 9, the suppression effect of each algorithm increase with the increase of the total cost of suppression nodes. However, the ACO-GE algorithm has the highest suppression effect under the same cost. At the same time, the ACO-GE algorithm has the lowest cost within the same suppression effect. For example, in the karate data set, when the suppression set cost is 2.0, the suppression effects of the Random, DegreeCentrality, PageRank, K-core, and CI algorithms are 18 nodes, 16 nodes, 15 nodes, 19 nodes, and 16 nodes, respectively, while the suppression effect of the ACO-GE algorithm is 21 nodes. When the suppression effect is 21, the cost of the Random, DegreeCentrality, PageRank, K-core, and CI algorithms are 3.0, 4.0, 4.5, 2.5, and 4.0, respectively, while the cost of the ACO-GE algorithm is only 2.0.

In Random algorithm, the total cost of restraining nodes is directly proportional to the average degree of network nodes, so the total cost increases linearly with the increase of the number of restraining nodes. Therefore, in Fig. 10, we only compare the other five algorithms. It can be seen from the figure that the total cost of DegreeCentrality algorithm and PageRank algorithm is similar and higher than that of ACO-GE algorithm. When selecting the suppression set, each algorithm tends to select the nodes with high cost first, so the slope of the curve shows a decreasing trend. Because the degree difference of each node in synthetic data set and football data set is small, and the cost difference of each node is small, the total cost of the five algorithms is almost linear.

Overall, the cost per node selected by the Random algorithm is approximately equal to the average cost of the network nodes. DegreeCentrality algorithm often selects the node with the largest cost, so its cost performance is low. The PageRank algorithm does not deliberately choose high-degree nodes, and its performance is higher than that of the random and DegreeCentrality algorithm. The high scalability of the CI algorithm is able to find the most influence nodes in the big data social networks. The K-core algorithm is a layered operation to determine the importance of the nodes in the network. Compared with traditional algorithms that do not consider the cost effectiveness of nodes, the ACO-GE algorithm uses the positive feedback mechanism of the ant colony algorithm to avoid selecting nodes exclusively according to their degree of performance. Therefore, the algorithm has the best combined effect in selecting nodes with limited cost.

5.5. Comparative analysis of community blocking effects of algorithms

Finally, we also compare the blocking effects of different algorithms in the network. Because many real networks tend to have community structure, the performance of different suppression algorithms for negative information interception determines whether the negative influence can spread from the source community to other communities. Therefore, we also need to evaluate the suppression effect of different algorithms in different communities. For example, during the epidemic period, if

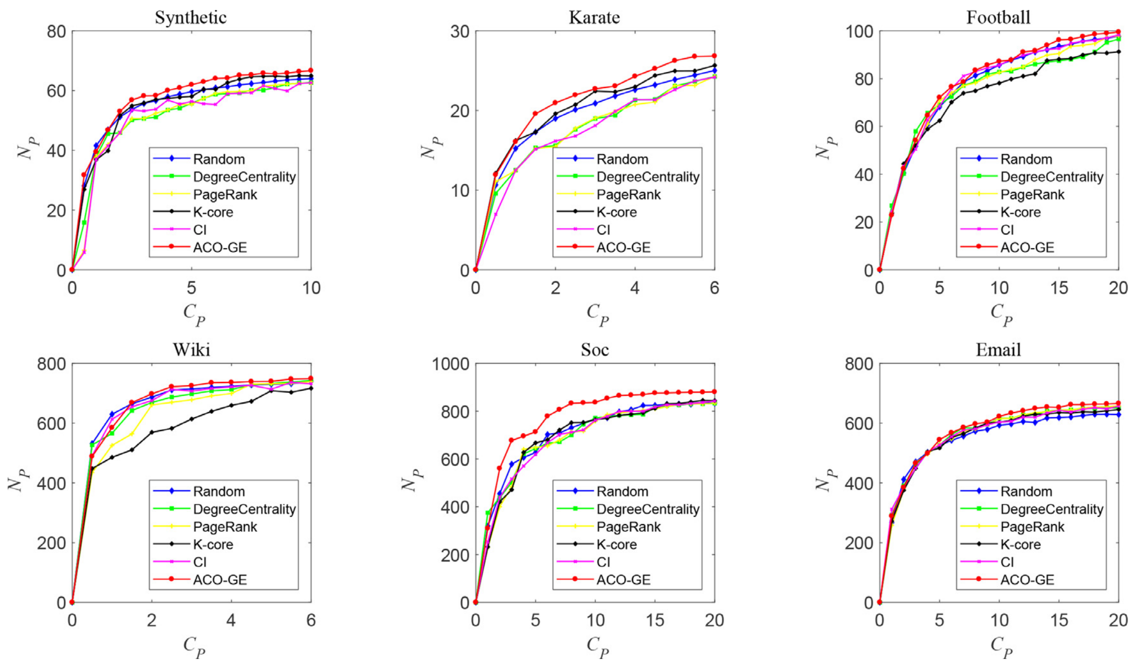

Fig. 9. The trend of the maximum number of nodes that can be affected by the suppression nodes selected by different algorithms changes with the increase of cost.

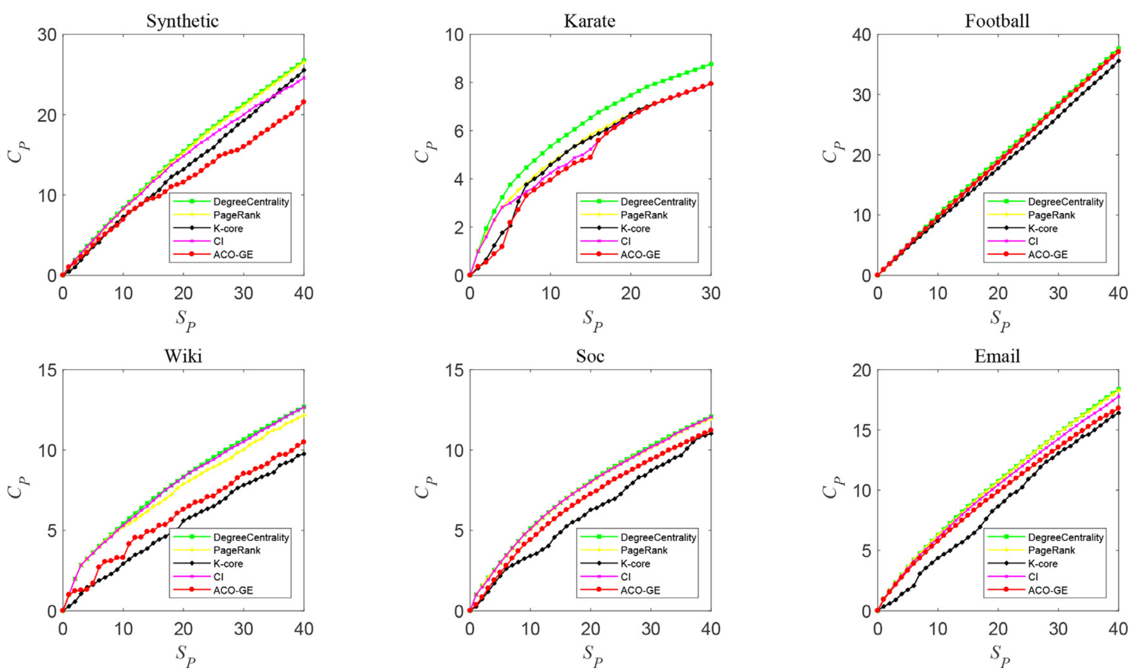

Fig. 10. The trend of the cost of different algorithms changes with the increase of the number of suppression nodes.

the epidemic spread can be limited to a few communities, medical resources can be used intensively, so as to save a lot of medical resources and reduce the burden of community workers. In detail, if there are two different suppression methods, after blocking the spread of negative influence, although the number of nodes affected by negative influence is similar, the nodes affected by negative influence of the first algorithm are mainly scattered in different communities, while the nodes affected by negative influence of the second algorithm are mainly concentrated in individual communities, Then, we think that the second algorithm has better community blocking effect, because the second method can avoid the further radiation and diffusion of negative influence in the network in the next time. The total number of negative nodes in the network reflects the algorithm's blocking effect on negative influence, while the number of communities with negative nodes in the network reflects the algorithm's community blocking effect.

For each network, we randomly select 4% of nodes from a specific community as the initial negative seed set and then compare the effect of different suppression algorithms and the propagation of negative influence without taking any measures. In order to reduce the experimental error, each algorithm performs 1000 times, and the average value is taken as the expected result. From Fig. 8, we can see that the number of nodes affected by each algorithm is less when the k of the suppression set is too small, and the number of nodes affected by each algorithm tends to be the same when the k is too large. Therefore, in

Table 3
The blocking effect results of various algorithms in different data sets.

Data sets	Method	NM	NMR	NO	NOR	NIC	NA	NAR
Synthetic	No measures	19.33	87.86%	43.97	67.65%	2.56	63.30	72.76%
	Random	14.46	65.73%	3.38	5.21%	1.77	17.84	20.51%
	DegreeCentrality	13.53	61.50%	0.59	0.90%	1.21	14.18	16.30%
	Pagerank	13.78	62.64%	0.38	0.58%	1.15	14.16	16.28%
	CI	15.00	68.18%	0.48	0.73%	1.29	15.48	17.79%
	K-core	14.81	67.32%	3.29	5.05%	1.71	18.10	20.80%
	ACO-GE	**8.05**	**36.59%**	**0.35**	**0.54%**	**1.09**	**8.40**	**9.66%**
Karate	No measures	8.61	71.75%	16.37	74.41%	3.83	24.98	73.47%
	Random	4.43	36.92%	3.16	14.36%	2.69	7.59	23.32%
	DegreeCentrality	3.41	28.42%	1.86	8.45%	1.85	5.27	15.50%
	Pagerank	3.28	27.33%	1.58	7.18%	1.79	4.86	14.29%
	CI	4.51	37.58%	1.36	6.18%	2.13	5.87	17.26%
	K-core	**2.99**	**24.92%**	3.97	18.05%	2.33	6.96	20.47%
	ACO-GE	3.67	30.58%	**0.26**	**1.18%**	**1.26**	**3.93**	**11.56%**
Football	No measures	15.10	88.82%	80.91	82.56%	11.84	96.01	83.49%
	Random	11.01	64.76%	20.69	21.11%	8.63	31.70	27.57%
	DegreeCentrality	**8.68**	**51.06%**	15.13	15.44%	6.92	23.81	20.70%
	Pagerank	9.07	53.35%	15.20	15.51%	7.03	24.27	21.10%
	CI	9.73	57.24%	14.24	14.53%	7.15	23.97	20.84%
	K-core	10.31	60.65%	26.5	27.04%	8.89	36.81	32.01%
	ACO-GE	11.62	68.35%	**12.15**	**12.40%**	**6.54**	**23.77**	**20.67%**
Wiki	No measures	188.26	37.80%	605.15	31.73%	16.69	793.41	32.99%
	Random	84.47	16.96%	316.77	13.17%	13.92	401.24	16.68%
	DegreeCentrality	36.48	7.33%	35.91	1.88%	8.80	72.39	3.01%
	Pagerank	29.38	5.89%	38.46	2.02%	8.92	65.84	2.73%
	CI	26.77	5.38%	42.15	2.21%	9.85	68.92	2.87%
	K-core	53.23	10.69%	139.54	7.32%	11.85	192.78	8.02%
	ACO-GE	**25.61**	**5.14%**	**34.28**	**1.79%**	**8.52**	**59.89**	**2.49%**
Soc	No measures	104.57	18.91%	871.14	26.97%	5.71	975.71	25.79%
	Random	61.05	11.06%	423.57	13.11%	5.18	484.62	12.81%
	DegreeCentrality	11.85	2.13%	49.57	1.53%	3.97	61.42	1.62%
	Pagerank	14.43	2.61%	52.14	1.62%	3.86	66.57	1.76%
	CI	17.13	3.16%	39.16	1.22%	4.29	56.29	1.50%
	K-core	13.28	2.40%	55.17	1.71%	4.14	68.42	1.80%
	ACO-GE	**9.41**	**1.71%**	**33.58**	**1.02%**	**3.48**	**42.99**	**1.14%**
Email	No measures	197.23	55.56%	347.69	53.49%	5.83	544.92	54.22%
	Random	52.69	14.84%	73.15	11.25%	5.38	125.85	12.52%
	DegreeCentrality	11.31	3.19%	9.38	1.44%	2.87	20.69	2.06%
	Pagerank	**9.15**	**2.56%**	9.77	1.50%	3.03	18.92	1.88%
	CI	10.92	3.08%	8.38	1.29%	3.15	19.31	1.92%
	K-core	11.69	3.29%	10.85	1.67%	3.23	22.54	2.24%
	ACO-GE	9.38	2.64%	**7.69**	**1.18%**	**2.39**	**17.08**	**1.70%**

this experiment, the k of the suppression set of the Wiki data set is 5, and the k of the suppression set of the other data sets is 10. The blocking effects of various algorithms are shown in Table 3. Among them, NM and NMR are the number and proportion of affected nodes in the community where the initial negative node is located, respectively; NO and NOR are the number and proportion of nodes affected by negative influence in other communities respectively; NIC is the number of communities containing negative nodes; NA and NAR are the number and proportion of nodes affected by negative influence in the whole network, respectively.

Table 3 shows that the ACO-GE algorithm has the best inter-community blocking ability, the DegreeCentrality algorithm and the PageRank algorithm have similar effects, and the Random algorithm has the worst effect. The suppression set selected by different algorithms affects different communities. The ACO-GE algorithm can effectively prevent the spread of negative influence from the source community of negative influence to other healthy communities. For example, in the karate data set, the suppression effect of ACO-GE in the source community of negative influence is less than the DegreeCentrality algorithm and PageRank algorithm, but the suppression effect in other communities is better, which effectively prevents the spread of negative influence in the network.

In summary, the ACO-GE algorithm takes into account the global topology of the network, and has the best community blocking effect while suppressing negative influence nodes. Therefore, the comprehensive performance of the ACO-GE algorithm is better than the other five algorithms.

6. Conclusion

Influence maximization of complex networks is an important research topic in social network analysis. When the principle of network communication is unknown, how to effectively suppress the negative impact is particularly important. Swarm intelligence is a simulation of colony behaviors such as ants and birds in the biological world, as well as a process of foraging between groups by means of cooperation. Individuals in the group search for the direction of the path through learning from themselves and between individuals, therefore, they can form a more powerful overall ability.

This paper combines the spread process of influence with ant colony foraging behavior, and simulates the spread of negative influence in complex networks through ant path selection. In the process of mutual cooperation, these individuals build the swarm intelligence through their relationship so that they can seek the optimal solution to complex problems based on ant colony intelligence in cyberspace, that is, finding a group of high-value and Low-cost suppression node. In the process of model design, we map the original network into a low-dimensional vector through the graph embedding technology, capture the topology of the original network and the relationship between nodes, get more potential information between nodes, and use it as the metric for the heuristic factor of ant colony algorithm. Experiments show that this algorithm can effectively suppress the spread of negative influence in the whole network with limited cost and in the network with community structure. The suppression effect is community blocking, which can concentrate on the spread of negative influence in individual communities and avoid the wireless increase of community coverage of negative influence. Therefore, the research of this paper can not only enrich the theoretical research results of influence maximization, but also play an important role in the fields of network topology analysis, epidemic prevention and control, rumor propagation and so on.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Bo-Lun Chen: Conceptualization, Methodology, Validation, Writing – original draft, Funding acquisition. **Wen-Xin Jiang:** Conceptualization, Investigation, Data curation, Software, Writing – original draft. **Yong-Tao Yu:** Conceptualization, Methodology, Validation, Writing – original draft. **Lei Zhou:** Conceptualization, Software, Writing – review & editing. **Claudio J. Tessone:** Conceptualization, Writing – review & editing.

Acknowledgment

This research is supported in part by the National Natural Science Foundation of China under grant No. 61602202, the Natural Science Foundation of Jiangsu Province under contract No. BK20160428, and the Natural Science Foundation of Education Department of Jiangsu Province under contract No. 20KJA520008. Six talent peaks project in Jiangsu Province (Grant No.XYDXX-034) and China Scholarship Council also supports this work.

References

[1] W. Chen, T. Lin, Z. Tan, M. Zhao, X. Zhou, Robust influence maximization, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 795–804.
[2] Y. Li, J. Fan, Y. Wang, K.-L. Tan, Influence maximization on social graphs: a survey, IEEE Trans. Knowl. Data Eng. 30 (10) (2018) 1852–1872.
[3] N. Barbieri, F. Bonchi, Influence Maximization with Viral Product Design, US Patent 10,546,308, 2020.
[4] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (7563) (2015) 65–68.
[5] S. Banerjee, M. Jenamani, D.K. Pratihar, A survey on influence maximization in a social network, Knowl. Inf. Syst. 62 (9) (2020) 3417–3455.
[6] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016) 1–63.
[7] D.S. Maynard, T.W. Crowther, M.A. Bradford, Competitive network determines the direction of the diversity–function relationship, Proc. Natl. Acad. Sci. 114 (43) (2017) 11464–11469.
[8] L. Sun, W. Huang, P.S. Yu, W. Chen, Multi-round influence maximization, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2249–2258.
[9] A. Caliò, R. Interdonato, C. Pulice, A. Tagarelli, Topology-driven diversity for targeted influence maximization with application to user engagement in social networks, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2421–2434.
[10] S. Chakraborty, S. Stein, M. Brede, A. Swami, G. de Mel, V. Restocchi, Competitive influence maximisation using voting dynamics, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019, pp. 978–985.
[11] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.
[12] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.
[13] X. He, G. Song, W. Chen, Q. Jiang, Influence blocking maximization in social networks under the competitive linear threshold model, in: Proceedings of the 2012 SIAM International Conference on Data Mining, SIAM, 2012, pp. 463–474.
[14] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 420–429.
[15] G. Tong, W. Wu, L. Guo, D. Li, C. Liu, B. Liu, D.-Z. Du, An efficient randomized algorithm for rumor blocking in online social networks, IEEE Trans. Netw. Sci. Eng. 7 (2) (2017) 845–854.
[16] P. Wu, L. Pan, Scalable influence blocking maximization in social networks under competitive independent cascade models, Comput. Netw. 123 (2017) 38–50.
[17] C.V. Pham, H.M. Dinh, H.D. Nguyen, H.T. Dang, H.X. Hoang, Limiting the spread of epidemics within time constraint on online social networks, in: Proceedings of the Eighth International Symposium on Information and Communication Technology, 2017, pp. 262–269.
[18] E.B. Khalil, B. Dilkina, L. Song, Scalable diffusion-aware optimization of network topology, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1226–1235.
[19] Y. Zhang, B.A. Prakash, Data-aware vaccine allocation over large networks, ACM Trans. Knowl. Discov. Data (TKDD) 10 (2) (2015) 1–32.
[20] N. Arazkhani, M.R. Meybodi, A. Rezvanian, Influence blocking maximization in social network using centrality measures, in: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), IEEE, 2019, pp. 492–497.
[21] C.-L. Lee, C.-E. Sung, H.-S. Ma, J.-W. Huang, Idr: positive influence maximization and negative influence minimization under competitive linear threshold model, in: 2019 20th IEEE International Conference on Mobile Data Management (MDM), IEEE, 2019, pp. 501–506.
[22] S. Peng, M. Wu, G. Wang, S. Yu, Containing smartphone worm propagation with an influence maximization algorithm, Comput. Netw. 74 (2014) 103–113.
[23] C.J. Kuhlman, G. Tuli, S. Swarup, M.V. Marathe, S. Ravi, Blocking simple and complex contagion by edge removal, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 399–408.
[24] L. Xue, P. Zhang, A. Zeng, Maximizing spreading in complex networks with risk in node activation, Inf. Sci. 586 (2022) 1–23.
[25] N. Arazkhani, M.R. Meybodi, A. Rezvanian, An efficient algorithm for influence blocking maximization based on community detection, in: 2019 5th International Conference on Web Research (ICWR), IEEE, 2019, pp. 258–263.
[26] J. Lv, B. Yang, Z. Yang, W. Zhang, A community-based algorithm for influence blocking maximization in social networks, Cluster Comput. 22 (3) (2019) 5587–5602.
[27] B. Wang, G. Chen, L. Fu, L. Song, X. Wang, DRIMUX: dynamic rumor influence minimization with user experience in social networks, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2168–2181.
[28] W. Zhu, W. Yang, S. Xuan, D. Man, W. Wang, X. Du, Location-aware influence blocking maximization in social networks, IEEE Access 6 (2018) 61462–61477.
[29] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, Y. Bi, Least cost rumor blocking in social networks, in: 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 2013, pp. 540–549.
[30] M. Gong, C. Song, C. Duan, L. Ma, B. Shen, An efficient memetic algorithm for influence maximization in social networks, IEEE Comput. Intell. Mag. 11 (3) (2016) 22–33.
[31] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, F. Xia, Community-diversified influence maximization in social networks, Inf. Syst. 92 (2020) 101522.
[32] D. Sivaganesan, Novel influence maximization algorithm for social network behavior management, J. ISMAC 3 (01) (2021) 60–68.
[33] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[34] C. McCormick, Word2vec tutorial-the skip-gram model, Apr-2016.[Online]. Available: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model(2016).
[35] Q. Liu, B. Xiang, N.J. Yuan, E. Chen, H. Xiong, Y. Zheng, Y. Yang, An influence propagation view of pagerank, ACM Trans. Knowl. Discov. Data (TKDD) 11 (3) (2017) 1–30.
[36] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893.