



Influence maximization in mobile social networks based on RWP-CELF

Zhenyu Xu^{1,2} · Xinxin Zhang³ · Mingzhi Chen^{1,2} · Li Xu^{1,2}

Received: 8 March 2023 / Accepted: 19 February 2024 / Published online: 21 March 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2024

Abstract

Influence maximization (IM) problem for messages propagation is an important topic in mobile social networks. The success of the spreading process depends on the mechanism for selection of the influential user. Beside selection of influential users, the computation and running time should be considered in this mechanism to ensure the accuracy and efficient. In this paper, considering that the overhead of exact computation varies nonlinearly with fluctuations in data size, random algorithm with smoother complexity change was designed to solve the IM problem in combination with greedy algorithm. Firstly, we proposed a method named two-hop neighbor network influence estimator to evaluate the influence of all nodes in the two-hop neighbor network. Then, we developed a novel greedy algorithm, the random walk probability cost-effective with lazy-forward (RWP-CELF) algorithm by modifying cost-effective with lazy-forward (CELF) with random algorithm, which uses 25–50 orders of magnitude less time than the state-of-the-art algorithms. We compared the influence spread effect of RWP-CELF on real datasets with a theoretically proven algorithm that is guaranteed to be approximately optimal. Experiments show that the spread effect of RWP-CELF is comparable to this algorithm, and the running time is much lower than this algorithm.

Keywords Influence maximization · Mobile social network · Two-hop neighbor network influence estimator · Random algorithm · Greedy algorithm

Mathematics Subject Classification 9103D

1 Introduction

The continuous combination of social networks and mobile internet has prompted the emergence and development of mobile social networks (MSNs), which enable people to spread information and opinions more quickly and widely [1]. As a tool for user communication, a piece of information in mobile social networks can

spread quickly through “word of mouth” among friends, and this diffusion phenomenon has been found to have many applications, such as viral marketing [2], rumor control [3], etc. Viral marketing means that the companies select a group of high-influence users in the network to spread product information in the circle of friends, so that they can achieve the purpose of product promotion in the entire network at a lower cost [4]. The spread of rumors through online networks not only threatens public safety but also results in loss of financial property, one of the ways to achieve rumor control is to select some users in the network to spread an anti-rumor message to curb the spread of rumors [5]. Researchers put forward the influence maximization (IM) problem to study this kind of communication phenomenon, which has attracted widespread attention. [6].

The goal of IM problem is to find a set of influence seed nodes with a size of k , and spread the information to the entire network as much as possible under a specific propagation model through these k seed nodes. Kempe et al. [7] first regarded the influence maximization problem as a discrete optimization problem, and proved that it is an NP-hard problem. With the development of research, many other related problems based on IM problems have been proposed [8], such as Budgeted Influence Maximization Problem (BIM Problem) [9], Weighted Target Set Selection Problem (WTSS Problem) [10], etc. One crucial challenge in influence maximization is to quickly find the seed node set while ensuring the effective spread of influence. Many greedy and heuristic algorithms have been proposed to solve the influence maximization problem, but they often fail to guarantee fast search for the set of seed nodes and great influence spread at the same time [8]. We focus on finding seed node set more efficiently while ensuring effective influence spread. The main contributions are listed as follows:

- We propose the two-hop neighbor network influence estimator (TNNIE), the method can evaluate the node’s influence in its two-hop neighbor network accurately and quickly, and we theoretically proved the feasibility of the algorithm.
- We design a novel random algorithm named random walk probability (RWP) to evaluate the influence of candidate seed nodes and obtain their set of neighbor nodes. Based on this, we also develop random walk probability cost-effective with lazy-forward (RWP-CELF), a greedy algorithm, to ensure that the problem of overlapping influence between nodes is avoided, and quickly find the set of seed node.
- The experiments results of six real datasets show that our algorithm runs 20–50 times faster than state-of-the-art algorithms and has comparable influence spread effects to theoretically proven algorithm that are guaranteed to be approximately optimal.

The rest of this paper is organized as follows. We introduce two diffusion models commonly used to achieve influence maximization and some related work to IM problem in Sect. 2. In Sect. 3, We introduce how to extract mobile social networks into a graph to solve the problem of maximizing influence and show the methods we proposed to solve IM problem. In Sect. 4, we demonstrate the effectiveness and efficiency of our proposed methods through experiments. Finally, Sect. 5 concludes this paper.

2 Related work

In this section, we first introduce two commonly used diffusion models and their properties, and then introduce some methods to solve influence maximization problem.

2.1 Influence diffusion models

The spread of influence is an important evaluation index of the influence maximization problem, which is demonstrated by simulation of the diffusion models. Here we introduce the two most widely used models: Independent Cascade Model (IC) and Linear Threshold Model (LT) [7].

Nodes are divided into two states in both models: active and inactive. In the IC model, when a node is activated at time t , it can only try to activate the inactive neighbor nodes at time $t + 1$, the probability of success is p , this behavior is independent. In the LT model, whether a node can change from an unactivated state to an active state depends on the activated in-degree nodes. The difficulty of each node activation is randomly generated, each node has an activation threshold $\theta \in (0, 1)$, and the behavior of the activated node is not independent. For an inactive node v , when the sum of the in-degree node's influence on it is greater than θ , node v becomes the active state. Our algorithm focuses on influence maximization under the IC model.

In addition to the two commonly used models mentioned above, many variant propagation models based on these two models have been proposed [11]. For example, Zhu et al. [12] proposed a classification metric method called SpreadRank and designed the Continuous Time Markov Chain Independent Cascade Model (CTMC-ICM) based on the IC model. Srivastava et al. [13] designed a new model based on the influence of nodes in the local network, taking into account social behavior among users and some other factors. Baghmolaei et al. [14] first introduced trust mechanism into the IC model and proposed a trust based latency aware independent cascade (TLIC) model. Wang et al. [15] extended the LT model and proposed the Linear Threshold with multi level Attention (LT-MLA) model by determining the state of nodes through various attitudes.

2.2 Presented approaches

The existing solutions for IM problem can be divided into the following three categories: greedy algorithms, heuristic algorithms and other algorithms.

2.2.1 Greedy algorithms

Kempe et al. [7] firstly proposed a hill-climbing greedy algorithm: SimpleGreedy, and proved the errors bounded at $\left(1 - \frac{1}{e} - \varepsilon\right)$ to the optimal solution approximately. However, although the SimpleGreedy algorithm can guarantee high effectiveness, it takes a lot of time, and as the network scale becomes larger, its time consumption

becomes larger. To solve it, many methods were proposed to reduce the time consumed while ensuring the effectiveness of SimpleGreedy.

According to the submodularity of the diffusion model, Leskovec et al. [16] used *Lazy Evaluation* to reduce the number of node influence evaluations and proposed Cost-Effective with Lazy-Forward (CELF) to improve the efficiency of the SimpleGreedy algorithm up to 700 times, CELF guarantees to achieve at least a fraction $\frac{1}{2} \left(1 - \frac{1}{e} - \varepsilon \right)$ of the optimal solution. Goyal et al. [17] proposed CELF++ based on CELF algorithm, which further improves the efficiency, but it is still not suitable for large-scale network scenarios. Chen et al. [18] proposed a greedy-based algorithm: NewGreedy, which makes the structure of the network simpler by removing the edges that are not involved in the propagation of the network, thereby reducing the running time of the SimpleGreedy algorithm. Kundu and Pal [19] proposed a deprecation-based greedy strategy to realize the selection of seed nodes for large-scale social networks, and from the perspective of theoretical proof, it shows that the algorithm can correctly identify the node to be discarded for any monotonic and sub-module influence function. Shang et al. [20] proposed CoFIM for the problem of maximizing the influence of large-scale non-overlapping community networks, they assume that the influence diffusion of nodes can be divided into cross-community and intra-community. Lu et al. [21] proposed a CascadeDiscount algorithm, which estimates the marginal revenue of nodes by removing the influence loss of nodes on neighbors after evaluating the initial influence, and then selects the influence seed node set based on a greedy strategy.

2.2.2 Heuristic algorithms

To obtain influence seed node set quickly, scholars have proposed many heuristic algorithms based on centrality metrics. The simplest methods are to select the top k nodes of a given centrality metric, such as degree centrality [22], closeness centrality [22], betweenness centrality [23], etc., as the set of influence seed nodes. However, the influence nodes of these high centrality metrics may be aggregated, which leads to a possible overlap of influence ranges among the selected influence seed node set, thus making the algorithm ineffective. Chen et al. [18] proposed two heuristic algorithms: SingleDiscount and DegreeDiscount to select influence seed node set by considering the effect of the currently selected seed node on subsequent nodes. According to the experimental results, the DegreeDiscount algorithm shows a better performance than the SingleDegree algorithm. Gao et al. [24] proposed a new notion to evaluate social influence, which is measured by two indices based on local attributes and behavioral characteristics. Zhang et al. [25] proposed a scheme that integrates ITÖ algorithm into PSO algorithm to solve the problem of maximizing the influence in MSNs.

The heuristic algorithms can select the influential seed node set in a short time, but this kind of algorithm often has bad performance in the influence spread. The results of dealing with the problem of maximizing influence are not very satisfactory.

The meta-heuristic algorithms were found to be suitable for solving the problem of maximizing influence. By imitating the behavior of biological populations and the evolution process of some phenomena in physics, the seed node set is obtained. Jiang et al. [26] made the first attempt, they first proposed the EDV (Expected

Diffusion Value) influence evaluation function to evaluate the expected value of the spread of nodes within one-hop neighbors, and then through the process of optimizing the EDV function through the simulated annealing algorithm to select the node set, the experimental results show this algorithm can obtain an efficiency improvement that is 2–3 orders of magnitude higher than that of the traditional greedy algorithm. Based on EDV, Cui et al. [27] proposed an evolutionary algorithm named degree-descending search evolution (DDSE) based on a degree-decreasing search strategy to select seed nodes. Gong et al. [28] proposed LIE (Local Influence Estimation) function to evaluate the influence expected value of seed node set in the two-hop neighbor network. Inspired by the efficient evolutionary mechanism based on swarm intelligence, Tang et al. [29] proposed a discrete shuffled frog-leaping algorithm for the influence maximization problem.

2.2.3 Other algorithms

In addition to the above two types of methods, there are many other types of methods proposed to solve IM problem. For example,

Zhang et al. [30] conducted research on a small network, extracted network information to construct a transfer probability matrix, and used a clustering algorithm to divide the network into communities to identify influential nodes.

Shang et al. [31] designed a framework based on community structure for large-scale networks to solve the problem of maximizing influence. By dividing the propagation process into two stages, the influence of nodes in the network is evaluated from different stages. In addition, they also proposed a framework IMPC based on the multi-neighbor potential, and designed an objective function to approximate the influence of nodes in the network [32].

Farzaneh et al. [33] proposed the IMBC (Influence Maximization Based on Community Structure) algorithm, which abstracted communities as hypergraph nodes to prune the network, thereby reducing network computational overhead and improving efficiency.

Kim et al. [34] proposed an Independent Path Algorithm (IPA), which considers the probability that all paths between two nodes may have an impact, and assumed that the propagation process between each path exists independently of each other, through parallel computing Improve efficiency.

Liu et al. [35] also considered the possibility of influence transfer between nodes from the propagation path, and improved the computational efficiency by pruning the propagation path between nodes.

Although the greedy algorithm can show a good propagation effect, it has the problem of high complexity. As the network complexity increases and the network scale becomes larger, the time cost of this type of algorithm will become unacceptable; heuristic The meta-heuristic algorithm simply uses nodes with high influence as seed nodes, and does not take into account the overlapping influence between nodes; the meta-heuristic algorithm may fall into a local optimum, resulting in the selected seed nodes failing to achieve the optimal Excellent communication effect; community-based methods may not perform satisfactorily when faced with network situations without obvious community structures.

3 Influence maximization based on RWP-CELLF

In this section, we first introduce how to extract mobile social networks into a graph to solve the problem of maximizing influence and the overview of proposed methods. Then, our method, TNNIE, which is used to evaluate the node's initial influence, will be elaborated on, and we proved its rationality. Next, we describe how RWP algorithm evaluates the influence of candidate seed nodes. Finally, We explain how the RWP-CELLF algorithm avoids the influence overlap problem and the way it selects the set of seed nodes.

3.1 Extracted mobile social networks and overview of methods

Mobile social networks are often abstracted as graphs to solve the problem of maximizing influence. The structure of mobile social networks can be represented by a graph $G = (V, E)$. A node is used to represent an independent individual participating in the network, and V is a collection of these nodes. A certain relationship between two users in the network is abstracted as an edge between them, and E is used to represent the set of these edges. Nodes usually have two states in the network, active and inactive. The entire network is abstracted as a graph composed of user nodes and user relationships, and messages are transmitted between users through the edges in the network. The notations in this paper are shown in Table 1.

The overview of the proposed methods is shown in Fig. 1.

Firstly, the network is initialized, users are abstracted as nodes, and the relationship between users is regarded as edges. Then calculate the influence of all nodes according to the TNNIE algorithm and sort them in descending order. Select the node with the greatest influence in the ranking and assign it to u , add u to the seed node set S_k , and then calculate according to the RWP algorithm with u as the center to obtain the neighbor node set S and the probability set Q of these neighbor nodes being affected. According to Q and S , delete neighbor nodes in the network that can be highly influenced by u . Then calculate the top nodes with the highest influence among the remaining network nodes, select seed node to

Table 1 Notations

Notation	Definition
$G(V, E)$	Graph consisting of a set of nodes V and a set of edges E
v	Node v
d_u	The degree of node u
D_u	Normalized degree of node u
$\sigma^*(u)$	The influence of node u in its two-hop neighbor network
L	Random walk length
p	The propagation probability of each edge
p_r	The random walk probability
S	Node set
u	The most influential node among the remaining nodes
Q	The probability set of neighbor nodes being affected by influential node
k	The size of the seed node set
S_k	Seed set of size k

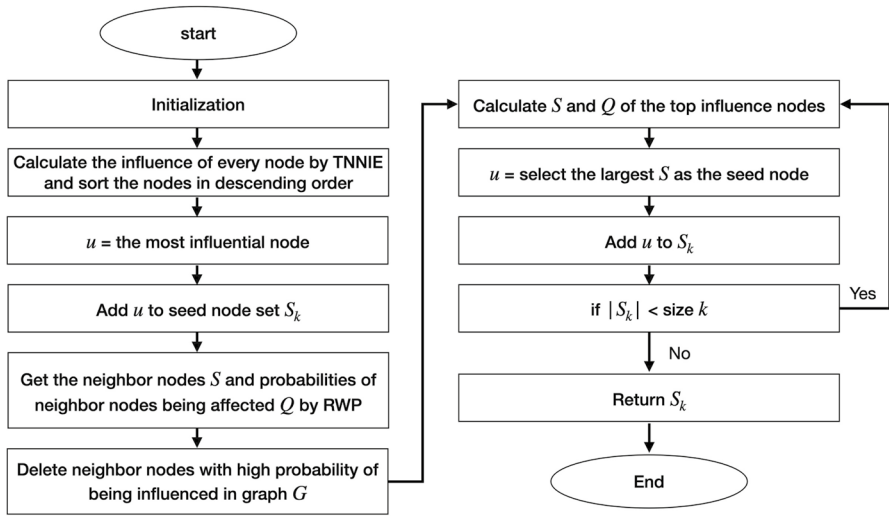


Fig. 1 Overview of the proposed methods

assign values to u according to the calculation results S of these nodes, add u to the seed node set S_k and delete neighbor nodes highly influenced by u . Repeat the above steps until the size of the seed node set is k , and finally output the seed node set.

3.2 Two-hop neighbor network influence estimator algorithm

Christakis and Fowler [36] considered that the spread of one’s influence is limited to a few local friends, and cannot spread to the entire network, named *Three Degree Theory*. Pei et al. [37] pointed out that the expected influence diffusion of nodes in the network depends on the influence of the second-order neighborhood of nodes. Following this, we propose Two-hop neighbor network influence estimator (TNNIE) to evaluate the influence of a node in the two-hop neighbor network and we demonstrate its reasonableness.

Lemma 1 *In the independent cascade model, give a graph $G(V,E)$, the influence of node u in its two-hop neighbor network is represented by $\sigma^*(u)$, when the propagation probability of each edge is a constant p , it can be defined as:*

$$\sigma^*(u) = 1 + p \cdot d_u + p^2 \cdot \sum_{v_i \in N_{(u)}^-} d_{v_i}$$

Proof In the IC model, the behavior of nodes to influence neighbor nodes is independent, so the expected influence spread of a node u in one-hop network can be computed as:

$$\sigma_1(u) = \sigma_0(u) + \sigma_1^*(u) = 1 + \sum_{v_i \in N_{(u)}^-} p_{u,v_i}$$

the p_{u,v_i} indicate the probability of node u activating its neighbor node v_i . Based on $\sigma_1(u)$, the expected influence spread of u in two-hop network can be computed as:

$$\sigma_2(u) = \sigma_0(u) + \sigma_1^*(u) + \sigma_2^*(u) = 1 + \sum_{v_i \in N^-(u)} p_{u,v_i} + \sum_{v_i \in N^-(u)} \sum_{v_j \in N^-(v_i)} p_{u,v_i} \cdot p_{v_i,v_j}$$

because the propagation probability of each edge is a constant p , so:

$$\begin{aligned} p_{u,v_i} &= p_{v_i,v_j} \\ \sum_{v_i \in N^-(u)} &= d_u \\ \sum_{v_j \in N^-(v_i)} &= d_{v_j} \end{aligned}$$

therefore

$$\sigma^*(u) = 1 + p \cdot d_u + p^2 \cdot \sum_{v_i \in N^-(u)} d_{v_i}$$

□

According to *Lemma 3.1*, the influence of a node in two-hop neighbor network can be calculated by Eqs. (1) and (2):

$$D_u = \frac{d_u}{d_{\max}} \tag{1}$$

$$\sigma^{2*}(u) = 1 + p \cdot D_u + p^2 \cdot \sum_{v_i \in N^-(u)} D_{v_i} \tag{2}$$

Normalize the degree of the node using the Eq. (2), the d_u indicate u 's degree and d_{\max} indicate the maximum node degree in the network.

3.3 Random walk probabiliy

The traditional greedy algorithm evaluates the influence of candidate seed nodes in the network through the Monte Carlo simulation method. The results obtained by this method are accurate, but this method takes a lot of time and is not applicable in large-scale networks. Therefore, we design the RWP algorithm, which is used for accurate and fast influence evaluation of candidate seed nodes when selecting seed node sets, and can obtain the neighbor node sets that may be influenced by candidate seed nodes to prepare for avoiding influence overlap.

Algorithm 1 Random Walk Probability (RWP)

Input: graph $G(V, E)$, candidate seed node u , walklength L , random walk probability p_r
Output: S_u stored the neighbor nodes of u , S_{inf} stored the neighbor nodes that may be influenced by u

- 1: Initialize: $S_o \leftarrow \phi$, $S \leftarrow \phi$, $S_{inf} \leftarrow \phi$
- 2: **for** $i = 1$ to $2 \cdot d_u$ **do**
- 3: $startnode = u$, $j = 0$
- 4: $S_o \leftarrow S_o \cup u$
- 5: **while** $j < L$ **do**
- 6: random select an neighboring node v from $startnode$
- 7: $S_o \leftarrow S_o \cup v$
- 8: $startnode = v$
- 9: $j++$
- 10: **end while**
- 11: **end for**
- 12: $S \leftarrow set(S_o)$
- 13: **for** each node n in S **do**
- 14: calculate the number of occurrences of n in S_o as Sum_n
- 15: $p_n = Sum_n / (2 \cdot d_u)$
- 16: **if** $p_n > p_r$ **then**
- 17: $S_{inf} \leftarrow n$
- 18: **end if**
- 19: **end for**
- 20: **return** S_{inf}

Random walk is often used to search for a node's neighbor nodes. Fu et al. [38] used random walk technology to obtain neighbors of each node, and gathered the vertices with similar neighbors into a community. Okuda et al. [39] proposed Restrained Random Walk to get a node's neighbors which are in the same community. Although random walk can obtain a node's neighbor node set. But whether one node can influence another node is a question of probability. As the number of hops increases, the influence of seed nodes will decrease, which means the probability of activating nodes will also decrease. Therefore, we design RWP algorithm, which calculates the probability of each neighbor node being walked by performing multiple random walks from as the probability of being influenced, and obtains the set of nodes consisting of neighboring nodes with high probability of being influenced.

The whole process of RWP is described in Algorithm 1. Where S_o is used to store the neighbor nodes traversed by each random walk (nodes that are repeatedly walked will be stored multiple times); S is used to store all the neighbor nodes that the random walk has gone through after de-duplication. $startnode$ indicated the node from which to start each random walk. Lines 2–11 represent a limited number of random walks from candidate seed node u , and the distance of each walk is L . In order to obtain the set of neighbor nodes as accurately as possible, the number of random walks is set to twice the degree of u . Lines 12–19 count the set of neighbor nodes that have been randomly walked from u after de-duplication, and calculate the probability that each neighbor node may randomly walk, nodes with random walk probability p_n higher than random walk probability p_r are stored in S_{inf} . Finally, the neighbor node set S_{inf} is returned, and the number of nodes in S_{inf} is the influence spread range of the candidate seed node u .

3.4 Random walk probability cost-effective with lazy-forward

CELf was proposed to solve the inefficiency of SimpleGreedy [16], it guarantees to achieve at least a fraction $\frac{1}{2} \left(1 - \frac{1}{e} - \varepsilon\right)$ of the optimal solution with a theoretically proven. It reduces the number of Monte Carlo simulations each time a seed node is picked according to submodularity, thus reducing the running time.

Although CELf can effectively maximize the influence, it still has problems such as inefficiency in finding the set of seed node, which is not suitable for large-scale networks. When the CELf algorithm selects the optimal seed node in each round, the candidate seed node and the already selected seed nodes are combined together to calculate the influence spread range through Monte Carlo simulation, which not only increases the redundancy of the selected seed node calculation, but also ignore the overlapping influence between nodes. Therefore, we develop RWP-CELf to select the seed node set more efficiently while ensuring that the problem of overlapping influence between nodes is avoided.

Algorithm 2 Random Walk Probability Cost-Effective with Lazy-Forward (RWP-CELf)

Input: graph $G(V, E)$, propagation probability p , random walk probability p_r , seed size k , walklength L

Output: a seed set S_k

```

1:  $S_k \leftarrow \phi$ 
2: for each  $v \in V$  do
3:    $v.inf = \sigma^{2*}(v)$ 
4: end for
5: sort the nodes in descending order based on  $v.inf$  to obtain the ranking list  $INF$ 
6: for  $i = 1$  to  $k$  do
7:    $S \leftarrow \phi, u = NULL, num = 0$ 
8:   if  $i = 1$  then
9:      $u = INF[num], S \leftarrow RWP(G, u, L, p_r)$ 
10:     $S_k = S_k \cup u$ 
11:    for each  $n$  in  $S$  do
12:      delete  $n$  from network  $G$ 
13:      delete  $n$  from list  $INF$ 
14:    end for
15:   else
16:     while  $u == NULL$  do
17:        $S_1, S_2 \leftarrow \phi$ 
18:        $u_1 = INF[num], S_1 \leftarrow RWP(G, u_1, L, p_r)$ 
19:        $u_2 = INF[num + 1], S_2 \leftarrow RWP(G, u_2, L, p_r)$ 
20:       if  $Num(S_1) > Num(S_2)$  then
21:          $S = S_1, u = u_1$ 
22:       repeat lines 11-14
23:     else
24:        $num = num + 1$ 
25:     end if
26:   end while
27:   end if
28: end for
29: return  $S_k$ 

```

In RWP-CELF, after each round of seed node selection, the neighbor nodes of the node will be deleted from the network. When selecting a seed node next time, candidate nodes are selected in the pruned network and their influence spread evaluation is calculated. In this way, the problem of overlapping influence between candidate nodes and already selected seed nodes is avoided. The most influential node calculated by TNNIE is added to the seed node set in the first iteration, and in the following iterations, the RWP algorithm is used to calculate only the possible influence spread of candidate nodes in the pruned network and compare the spread range (without considering the influence spread of the already selected seed nodes), redundant computation is avoided, and the optimal node is added to the seed node set.

The whole process of RWP-CELF is described in Algorithm 2. The S_k is used to store the seed nodes. Initialize S_k in step 1. Then, calculate every node's influence by TNNIE and sort the nodes in descending order based on their influence to obtain the ranking list INF (lines 2–5). Lines 6–29 select the seed node set of size k . Initialize each parameter in line 7, S is used to store the set of neighbor nodes that the candidate seed node may influence. u is used to store a seed node, and num represents the rank of the select candidate seed node in the list INF . In lines 9–14, the first node in INF is selected to store in u , and get the node's S through RWP, then add u to S_k . For each node in S , delete the node from the network G and list INF . Lines 16–26 select the remaining seed nodes. When u is $NULL$, select two nodes ranked num and $num + 1$ as candidate seed nodes, compare S_1 of node u_1 and S_2 of node u_2 , if $Num(S_1) > Num(S_2)$, repeat lines 11–14, otherwise $num = num + 1$, and this while loop.

4 Experimental evaluation and results

This section verifies the effectiveness of our proposed methods through simulation experiments. Our experiments are run on a Mac OS system with a 2.6 GHz six-core Intel Core i7 and 16 GB of RAM. The codes used in the experiment are all written in python and operated on the Pycharm platform.

4.1 Data sets

We demonstrate the performance of our propose methods by conducting experiments on six real-world datasets. And in order to verify the generality of the scheme, both directed networks and undirected networks are selected for experiments.

- AstroPh and CondMat datasets [40] are undirected networks formed by the collaborative relationship between the authors.
- Slashdot [41] is a undirected social network, which shows popular technology-related news.
- The Epinions [42] data set comes from the consumer review website Epinions.com, which contains the who-trust-whom relationship. On this website, each user can comment on a certain product online, while other users will express trust and distrust of the comments.

- The Eu-Email [40] network was created by European research institutes collecting email data. Nodes represent email addresses, and the behaviors of sending and receiving email information between nodes constitute a set of edges.
- Stanford [41] is a network graph obtained from the Stanford University website. Nodes are used to represent web pages, and links between web pages are represented by directed edges.

Some statistics of six datasets are shown in Table 1. $|V|$ represents the number of nodes, and $|E|$ represents the network's edges number. d_{avg} represents the average node degree. AC represents the average clustering coefficient. *Type* indicates whether the network is a directed graph or an undirected graph

4.2 Baseline algorithms

To prove the effectiveness and efficiency of our approach, Our algorithm will be contrasted with five state-of-the-art algorithms on six datasets. The parameters of these comparison algorithms are set according to the suggestions given in the original text. The propagation probability we set in the IC model is the same as most of the comparative literature [27–29].

- CELF [16]: A notable greedy-based algorithm that searches for influential node sets based on submodularity property and “lazy-forward” strategy. The Monte-Carlo simulation time is set to 10,000 to estimate nodes' marginal gain.
- SSA [43]: An optimal sampling framework based on the idea of reverse influence sampling. The parameters ϵ is set to 0.1, and δ is set to 0.1 and 0.01.
- DPSO [28]: An effective meta-heuristic algorithm. After a certain number of optimization iterations, k seed nodes with the best LIE adaptation value will be selected. The learning factors are set to 2, and the inertia weight ω is set to 0.8.
- DDSE [27]: An evolutionary algorithm based on a degree-decreasing search strategy to select seed nodes. The diversity of the population is set to 0.6, the mutation probability is set to 0.1, and the crossover probability cr will be set to 0.4.
- DSFLA [29]: A discrete shuffled frog-leaping algorithm based on network topology characteristics to identify influential nodes. The parameter setting pattern where $F = 100$, $m = 20$, $n = 5$, $q = 3 * n/4$ and $Iter = 30$.

4.3 RWP-CELF parameter settings

In the RWP-CELF, there are two key parameters: random walk probability p_r and walklength L . According to the literature [36, 37], we set the size of L to 3. We choose the best random walk probability p_r through experiments.

We select ca-AstroPh dataset to test the best random walk probability p_r . The experimental results are shown in Table 2. As the results, we can observe that when

p_r is set to 0.4, the maximum spread size can be obtained while ensuring time efficiency (Table 3).

4.4 Influence spread comparison

We evaluate the performance of the six algorithms on six datasets through experiments. The maximal evolutionary generation g_{max} of DSFLA, DPSO, and DDSE is set to 100. For DPSO, DDSE, DSFLA, and RWP-CELF, perform 1000 simulation experiments to obtain the average influence spread for comparison.

Figure 2 shows the influence spread performance of the six algorithms in six datasets with IC model propagation probability $p = 0.01$. From Fig. 2, we can conclude that RWP-CELF achieves satisfactory influence spread in all six real networks. In the AstroPh network and CondMat network, RWP-CELF has almost achieved the same influence spread as CELF. In the Slashdot network and the Epinions network, RWP-CELF shows the best influence spread when the number of seed nodes is small, and as the number of seed nodes increases, the influence spread in the Slashdot network is weaker than CELF, DPSO, and DSFLA, only weaker than CELF and DPSO in the Epinions network. In the Eu-Email network, RWP-CELF is slightly inferior to CELF and DPSO and is obviously stronger than the other three algorithms. In the Stanford network, the influence spread of RWP-CELF is almost the same as the best algorithm. We can also observe that with the increase of the number of seed nodes, the influence propagation range of RWP-CELF presents a smooth growth trend, indicating that our method follows the strategy of finding the node with the largest marginal gain as the seed node.

4.5 Running time comparison

We compare the running time of six algorithms to find 100 seed nodes in six networks to show the efficiency of our algorithm. Among them, the unit of the running time of CELF is minutes, while the others are seconds.

In Fig. 3, we can observe that the running time of DDSE, SSA, and RWP-CELF under most networks is almost the same, In the AstroPh network and CondMat network, DDSE, SSA, and RWP-CELF even take less than 10 s to identify the seed node set. But as shown in Fig. 1, we can observe that RWP-CELF is better than

Table 2 Network datasets

Network	V	E	d_{avg}	AC	Type
AstroPh	18772	198110	21.107	0.677	Undirected
CondMat	23133	186936	16.162	0.055	Undirected
Slashdot	77360	905468	23.409	0.087	Undirected
Epinions	75879	508837	13.412	0.261	Directed
Eu-Email	265214	420045	3.168	0.456	Directed
Stanford	281903	2312497	16.406	0.598	Directed

Table 3 Parameter setting comparison experiment on ca-AstroPh

Tests	p_r	Running time(s)	Spread size
Test1	0.1	3.23	510
Test2	0.2	3.52	514
Test3	0.3	3.72	517
Test4	0.4	3.58	525
Test5	0.5	3.80	514
Test6	0.6	3.62	511

DDSE, and SSA in influence spread on all networks, and in the CondeMat network, RWP-CELF take the shortest time. Although the CELF algorithm performs almost the best in all networks in terms of efficiency, it takes too much time. In the Slashdot network, it even spends 31 h to search for the seed nodes. Although the DPSO and DSFLA algorithms also have good performance in influence spread, they take a much longer time than RWP-CELF. For example, in Eu-Email and Stanford networks, RWP-CELF only needs more than ten seconds to search for the seed node set, while DPSO and DSFLA require hundreds or even thousands of seconds.

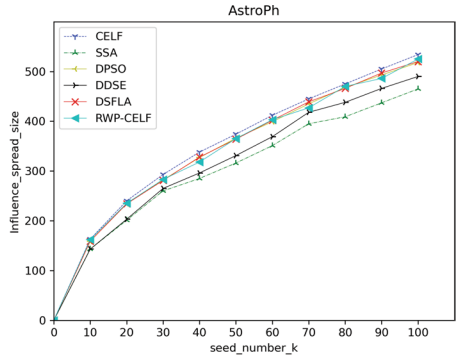
Although our algorithm has well performance in propagation efficiency and time efficiency, as RWP-CELF is an improved seed node selection algorithm based on CELF, which is an algorithm that improves greedy algorithms based on submodularity and only selects the current optimal target, there is a possibility that the seed node selected in each round of our algorithm may be in a local optimal situation. This leads to the fact that the selected seed node cannot be guaranteed to be the one that can achieve the highest actual marginal gain among all candidate nodes.

5 Conclusions

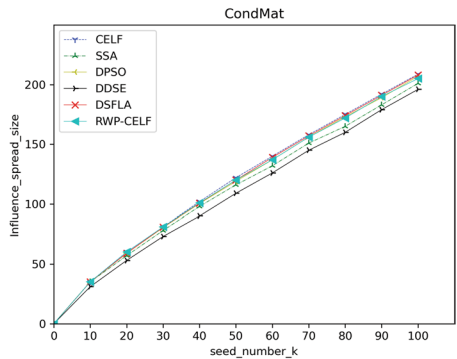
In this paper, considering that the overhead of exact computation varies nonlinearly with fluctuations in data size, random algorithm is designed to solve the IM problem in combination with greedy algorithm. Through experiments on six real data sets, it is shown that compared with the other five algorithms, our algorithm can achieve a short running time while ensuring the effectiveness of influence spread.

In future work, how to achieve higher global actual marginal gain when selecting seed nodes is one of our research directions, such as introducing optimization algorithms for improvement. And, how to ensure effective influence spread with better efficiency to solve the influence maximization problem in large-scale networks is the focus of our next research. Moreover, it is often not possible to obtain the complete network due to the high cost of obtaining the entire network topology. How to maximize the influence with the missing part of the network structure is also the next step of our research goal.

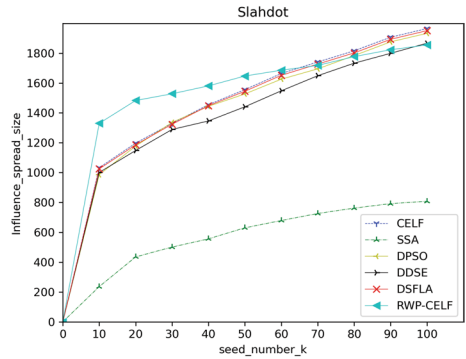
Fig. 2 Comparison of influence spread of the six algorithms on six datasets



(a) Astroph network

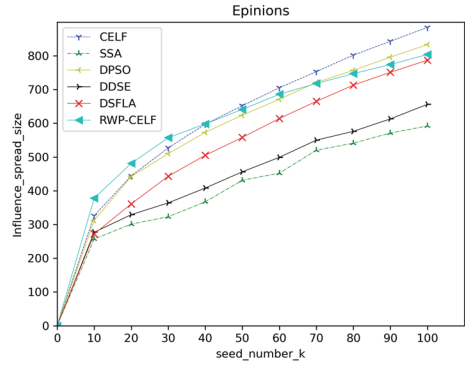


(b) CondMat network

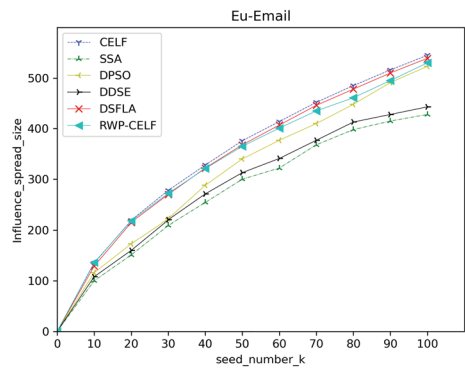


(c) Slashdot network

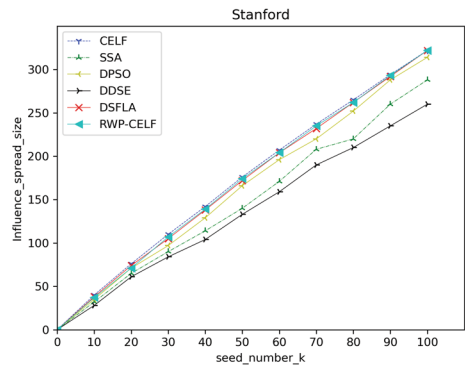
Fig. 2 (continued)



(d) Epinions network



(e) Eu-Email network



(f) Stanford network

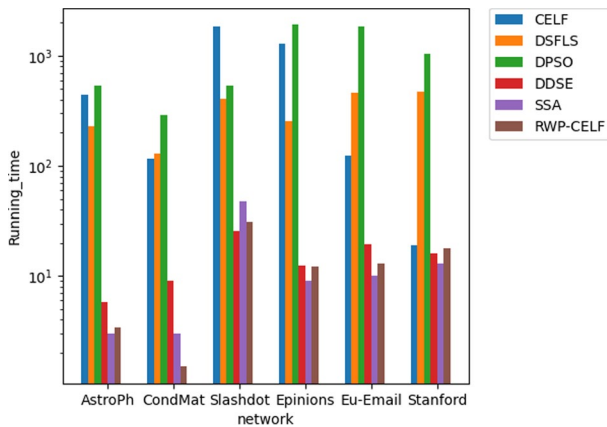


Fig. 3 Comparison on running time of the six algorithms on six datasets

Acknowledgements The authors would like to thank the National Natural Science Foundation of China No. U1905211, Science and technology projects in Fujian Province NOS. (2022G02003, 2021L3032), Fujian Provincial Department of Education Middle and Young People's Program No. JAT220814, Enterprise industry-university-research project DH-1565.

Declarations

Conflict of 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.

References

1. Hu X, Chu THS, Leung VCM, Ngai CH, Philippe K, Chan HCB (2015) A survey on mobile social networks: applications, platforms, system architectures, and future research directions. *IEEE Commun Surv Tutor* 17(3):1557–1581
2. Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1029–1038
3. Budak C, Agrawal D, Abbadi AE (2011) Limiting the spread of misinformation in social networks. In: *Proceedings of the 20th international conference on World wide web*, pp 665–674
4. Domingos P, Richardson M (2001) Mining the network value of customers. In: *Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 57–66
5. Zhang P, Bao Z, Niu Y, Zhang Y, Mo S, Geng F, Peng Z (2019) Proactive rumor control in online networks. *World Wide Web* 22(4):1799–1818
6. Li Y, Wang Y, Tan KL (2018) Influence maximization on social graphs: a survey. *IEEE Trans Knowl Data Eng* 30(10):1852–1872
7. Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: *Proceedings of the 9th international conference on knowledge discovery and data mining*, pp 137–146
8. Banerjee S, Jenamani M, Pratihari DK (2020) A survey on influence maximization in a social network. *Knowl Inf Syst* 62(9):3417–3455
9. Banerjee S, Jenamani M, Pratihari DK (2022) An approximate marginal spread computation approach for the budgeted influence maximization with delay. *Computing* 104:657–680

10. Raghavan S, Zhang R R (2015) Weighted target set selection on social networks, Technical report, Working paper, University of Maryland
11. Azaouzi M, Mnasria W, Romdhane LB (2021) New trends in influence maximization models. *Comput Sci Rev* 40:100393
12. Zhu T, Wang B, Wu B, Zhu C (2014) Maximizing the spread of influence ranking in social networks. *Inform ci* 278:535–544
13. Srivastava A, Chelms C, Prasanna VK (2014) Influence in social networks: a unified model? In: *Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining*, pp 451–454
14. Mohamadi-Baghmolaei R, Mozafari N, Hamzeh A (2015) Trust based latency aware influence maximization in social networks. *Eng Appl Artif Intell* 41:195–206
15. Wang F, Wang G, Xie D (2016) Maximizing the spread of positive influence under LT-MLA model. In: *Asia-Pacific services computing conference*, pp 450–463
16. Leskovec J, Krause A, Guestrin C, Faloutsos C, Vanbriesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp 420–429
17. Goyal A, Lu W, Lakshmanan LVS (2011) CELF++: optimizing the greedy algorithm for influence maximization in social networks. In: *Proceedings of the 20th international conference companion on World wide web*, pp 47–48
18. Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp 199–207
19. Kundu S, Pal SK (2015) Deprecation based greedy strategy for set selection in large scale social networks. *Inf Sci* 316:107–122
20. Shang J, Zhou S, Li X, Liu L, Wu H (2017) CoFIM: a community-based framework for influence maximization on large-scale networks. *Knowl-Based Syst* 117:88–100
21. Lu F, Zhang W, Shao L, Jiang X, Xu P, Jin H (2016) Scalable influence maximization under independent cascade model. *J Netw Comput Appl* 86:15–23
22. Freeman LC (1978) Centrality in social networks conceptual clarification. *Soc Netw* 1(3):215–239
23. Newman MJ (2005) A measure of betweenness centrality based on random walks. *Soc Netw* 27(1):39–54
24. Gao M, Xu L, Lin L, Huang Y, Zhang X (2020) Influence maximization based on activity degree in mobile social networks. *Concurr Comput Pract Exp*. <https://doi.org/10.1002/cpe.5677>
25. Zhang X, Xu L, Gao M (2020) An efficient influence maximization algorithm based on social relationship priority in mobile social networks. In: *International Symposium on Security and Privacy in Social Networks and Big Data*. Springer, pp 164–177
26. Jiang Q, Song G, Cong G, Wang Y, Si W, Xie K (2011) Simulated annealing based influence maximization in social networks. In: *Proceeding of the 25th AAAI conference on artificial intelligence*, pp 127–132
27. Cui L, Hu H, Yu S, Yan Q, Ming Z, Wen Z, Lu N (2018) DDSE: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *J Netw Comput Appl* 103:119–130
28. Gong M, Yan J, Shen B, Ma L, Cai Q (2016) Influence maximization in social networks based on discrete particle swarm optimization. *Inf Sci* 367–368(C):600–614
29. Tang J, Zhang R, Wang P, Zhao Z, Fan L, Liu X (2020) A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowl-Based Syst* 187:104833
30. Zhang X, Zhu J, Wang Q, Zhao H (2013) Identifying influential nodes in complex networks with community structure. *Knowl-Based Syst* 42:74–84
31. Shang J, Zhou S, Li X, Liu L, Wu H (2017) CoFIM: a community-based framework for influence maximization on large-scale networks. *Knowl-Based Syst* 117:88–100
32. Shang J, Wu H, Zhou S, Zhong J, Feng Y, Qiang B (2018) IMPC: influence maximization based on multi-neighbor potential in community networks. *Physica A* 512:1085–1103
33. Kazemzadeh F, Safaei AA, Mirzarezaee M, Afsharian S, Kosarirad H (2023) Determination of influential nodes based on the communities structure to maximize influence in social network. *Neurocomputing* 534:18–28

34. Kim J, Kim SK, Yu H (2013) Scalable and parallelizable processing of influence maximization for large-scale social networks. In: Proceedings of the IEEE 29th international conference on data engineering, pp 266–277
35. Liu B, Cong G, Zeng Y, Xu D, Chee YM (2014) Influence spreading path and its application to the time constrained social influence maximization problem and beyond. *IEEE Trans Knowl Data Eng* 26(8):1904–1917
36. Christakis NA, Fowler JH (2009) *Connected: the surprising power of our social networks and how they shape our lives*. Little, Brown, Boston
37. Pei S, Muchnik L, Andrade JS Jr, Zheng Z, Makse HA (2014) Searching for superspreaders of information in real-world social media. *Sci Rep* 4:5547
38. Fu X, Wang C, Wang Z, Ming Z (2013) Threshold random walks for community structure detection in complex networks. *J Softw* 8(2):286–295
39. Okuda M, Satoh S, Sato Y, Kidawara Y (2021) Community detection using restrained random-walk similarity. *IEEE Trans Pattern Anal Mach Intell* 43(1):89–103
40. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: densification and shrinking diameters. *ACM Trans Knowl Discov Data* 1(1):2-es
41. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. *Internet Math* 6(1):29–123
42. Richardson M (2003) Trust management for the semantic web. In: *Proceeding of international semantic web conference*, pp 351–368
43. Nguyen HT, Thai MT, Dinh TN (2016) Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: *Proceedings of the 2016 international conference on management of data*, pp 695–710

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Zhenyu Xu^{1,2} · Xinxin Zhang³ · Mingzhi Chen^{1,2} · Li Xu^{1,2}

✉ Li Xu
xuli@fjnu.edu.cn

Zhenyu Xu
1246233565@qq.com

Xinxin Zhang
zhangxx@haut.edu.cn

Mingzhi Chen
529015100@qq.com

¹ College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, China

² Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, Fujian, China

³ School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, China