

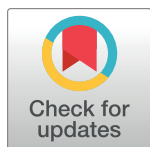
RESEARCH ARTICLE

# Maximizing multiple influences and fair seed allocation on multilayer social networks

Yu Chen<sup>1,2</sup>, Wei Wang<sup>1\*</sup>, Jinping Feng<sup>3</sup>, Ying Lu<sup>4</sup>, Xinqi Gong<sup>5\*</sup>

**1** School of Mathematics, Renmin University of China, Beijing, China, **2** School of Mathematics and Statistics, Minnan Normal University, Zhangzhou, Fujian province, China, **3** School of Mathematics and Statistics, Henan University, Kaifeng, Henan Province, China, **4** Faculty of Business and Economics, Hong Kong University, Hong Kong, China, **5** Institute for Mathematical Sciences, Renmin University of China, Beijing, China

\* [xinqigong@ruc.edu.cn](mailto:xinqigong@ruc.edu.cn) (XG); [wwai@ruc.edu.cn](mailto:wwai@ruc.edu.cn) (WW)



## OPEN ACCESS

**Citation:** Chen Y, Wang W, Feng J, Lu Y, Gong X (2020) Maximizing multiple influences and fair seed allocation on multilayer social networks. PLoS ONE 15(3): e0229201. <https://doi.org/10.1371/journal.pone.0229201>

**Editor:** Diego Raphael Amancio, University of Sao Paulo, BRAZIL

**Received:** July 7, 2019

**Accepted:** January 31, 2020

**Published:** March 12, 2020

**Copyright:** © 2020 Chen et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** Our code and data can be downloaded freely from <https://github.com/wingroy001/MMIC>.

**Funding:** This work was supported by the National Natural Science Foundation of China, under Grants [31670725, 91730301] obtained by Xinqi Gong, and the Educational Research Projects of Young and Middle-aged Teachers in Fujian Education Department, obtained by Yu Chen. These funding bodies did not play any role in the of study, the interpretation of data, or the writing of this manuscript.

## Abstract

The dissemination of information on networks involves many important practical issues, such as the spread and containment of rumors in social networks, the spread of infectious diseases among the population, commercial propaganda and promotion, the expansion of political influence and so on. One of the most important problems is the influence-maximization problem which is to find out  $k$  most influential nodes under a certain propagate mechanism. Since the problem was proposed in 2001, many works have focused on maximizing the influence in a single network. It is a NP-hard problem and the state-of-art algorithm IMM proposed by Youze Tang et al. achieves a ratio of 63.2% of the optimum with nearly linear time complexity. In recent years, there have been some works of maximizing influence on multilayer networks, either in the situation of single or multiple influences. But most of them study seed selection strategies to maximize their own influence from the perspective of participants. In fact, the problem from the perspective of network owners is also worthy of attention. Since network participants have not had access to all information of the network for reasons such as privacy protection and corporate interests, they may have access to only part of the social network. The owners of networks can get the whole picture of the networks, and they need not only to maximize the overall influence, but also to consider allocating seeds to their customers fairly, i.e., the Fair Seed Allocation (FSA) problem. As far as we know, FSA problem has been studied on a single network, but not on multilayer networks yet. From the perspective of network owners, we propose a multiple-influence diffusion model MMIC on multilayer networks and its FSA problem. Two solutions of FSA problem are given in this paper, and we prove theoretically that our seed allocation schemes are greedy. Subsequent experiments also validate the effectiveness of our approaches.

## Introduction

There are all kinds of complex networks in our life, such as social networks, the Internet of things, biological networks. The dynamic transmission of information in these networks is

**Competing interests:** On behalf of all authors, the corresponding author Xinqi Gong have declared that no competing interests exist. We adhere to PLOS ONE policies on sharing data and materials.

closely related to the networks' own topologies. Network scientists try to explain the dynamics of network by studying the spread of computer and mobile phone viruses [1][2], epidemic diseases [3], rumors [4] and so on. The authors attempt to understand all aspects of network dynamics, such as: the main body of spreading, the pattern of spreading, the carrier of spreading, the efficiency of spreading, and so on. In this paper, we focus on the spreading efficiency of nodes. We strive to find the most influential nodes in multilayer networks and allocate them to the networks' customers. Next, we will introduce some related works in three categories: 1. Influence maximization in a single network, 2. Influence maximization in multilayer networks, and 3. Fair seed allocation problem.

## Influence maximization in a single network

In recent years, with the rise of social media on the Internet, the communication between people has been enriched. Many social activities that used to be face-to-face are now available online, such as holding conferences, making friends, learning and counseling, etc. The growing demands for the content and form of social activities have led to the emergence of a large number of online social networks. More and more people like to express opinions and share information on online social networks such as Weibo, Facebook and Twitter. In these social networks, information is spread in the form of 'word-of-mouth'. Under this spreading mechanism, it is an important problem to find the most influential nodes in the social networks. Domingos et al. [5] first put forward the influence-maximization problem which is to find  $k$  most influential nodes in a social network, that is:

$$\operatorname{argmax}_{S \in V, |S|=k} \sigma(S) \quad (1)$$

where  $\sigma$  is the influence spread of  $k$  seeds in  $S$  and  $V$  is the node set.

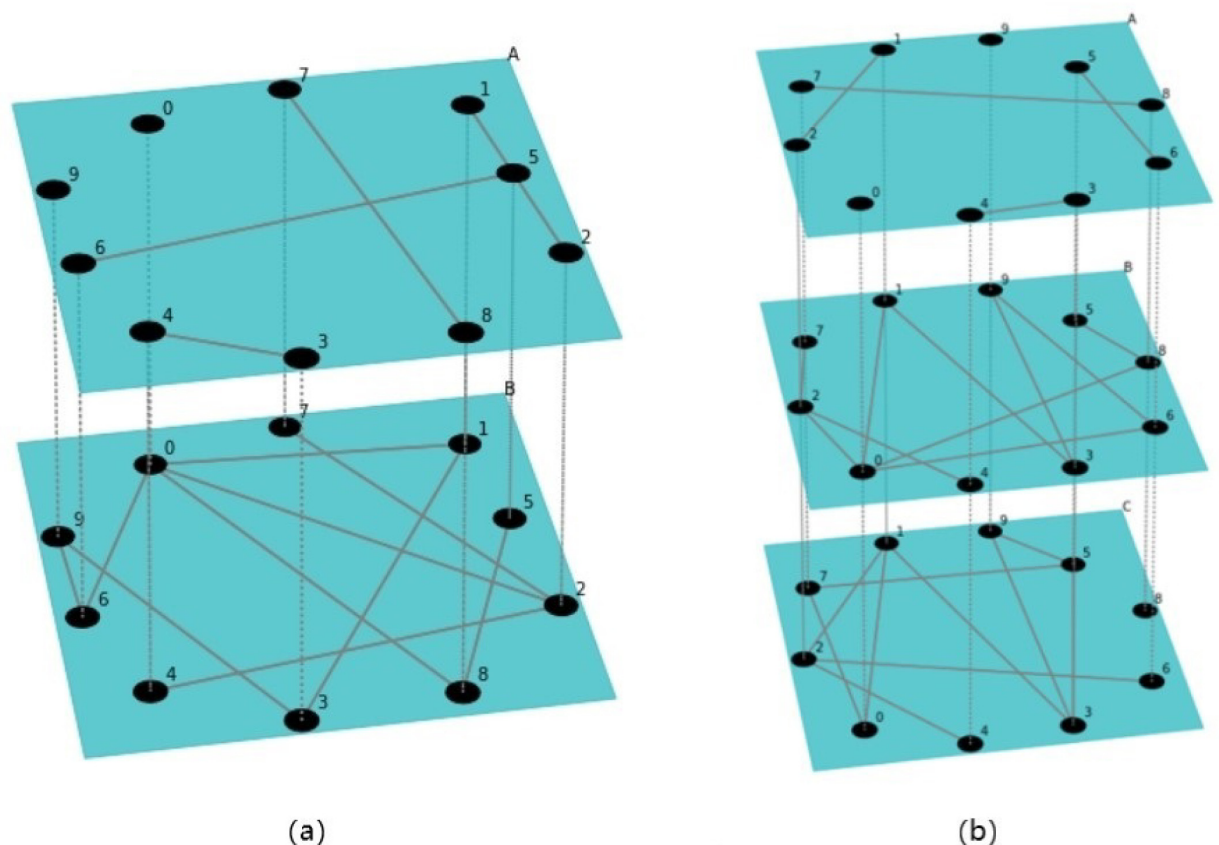
David Kempe et al. [6] proposed two basic models: independent cascade (IC) model and linear threshold (LT) model to describe the influence diffusion mechanism in a single network. However, both models are too simple to precisely reflect the diffusion mechanism on real social networks. To make up for this deficiency, many scholars focus on the improvement and modification of the models. Tim Carnes et al. [7] proposed two models for social network with competitive influences, and dealt with the influence-maximization problem from the follower's perspective. Budak C. et al. [8] proposed a Model named MCIC and studied the problem of the limitation of 'bad' influence. More models of multiple influences competing in a network are considered by researchers [9–14]. Furthermore, sale strategies for virus marketing are also studied [15]. Most previous works focused on the issue of influence propagation within a single network in the past decade. Since the influence-maximization problem is NP-hard [6], so far there is no algorithm of polynomial time complexity to solve the problem. All the algorithms proposed in the previous works are approximate algorithms, they focus on achieving an equilibrium between degree of approximation and time complexity of their algorithms. The state-of-art algorithm IMM [16] has approximately linear time complexity with theoretical guarantee of approximation.

Besides IC and LT model, there are some other network diffusion models, such as the Susceptible-Infectious-Susceptible model (SIS) [17, 18], the Susceptible-Infectious-Recovered model (SIR) [19, 20], the voter model (VM) [3] and the contact process (CP) [3], etc. However, IC and LT are more widely used in the study of influence maximization. The diffusion model used in this paper is IC model.

## Influence maximization in multilayer networks

Nowadays people act as entities in multiple social networks. It is normal for people to communicate and disseminate information in these networks synchronously. Therefore, compared with a single network, the influence-maximization problem in multilayer networks composed of multiple networks is more worth being studied. Fig 1 is an example of multilayer networks. Nodes marked with the same number represent the account of the same user in each layer of the network. Each user is called an entity, its account in each layer is called a representative of the entity.

In recent years, there have been successive works of maximizing influence in multilayer networks. Ibrahima Gaye et al. [21] proposed a centralized measurement method called 'Multi-Diffusion Degree' to select seeds in multilayer networks to maximize the influence. Li Guoliang et al. [22] used the maximum propagation path to approximate the influence between nodes and obtained several solutions of influence-maximization problem of multilayer networks. However, they did not introduce multiple influences into their models. The fact of the matter is that there is often not only one kind of influence spreading in real-world multilayer networks. For example, commercial advertising, the spread of public opinions, rumors and their suppression are all competing influences. Therefore, it is more meaningful to study the diffusion mechanism of multiple influences than that of single influence.



**Fig 1. Two examples of multilayer networks.** (a) has two layers: A and B. (b) has three layers A, B and C. Node A1 and node B1 are two representatives of entity '1', they are connected by a cross-layer edge (dash line).

<https://doi.org/10.1371/journal.pone.0229201.g001>

There are other works [23] on diffusion models of multilayer networks. They are mainly concerned with the spread of diseases, opinions and information among the population and how they can interact with each other. Ting Liu et al. [24] constructed a bilayer network, one is the contact network of epidemic spreading which adopts the Susceptible-Infected-Recovered (SIR) model to depict its spreading process, the other is the network of disease information which adopts IC model to depict its spreading process. Velásquez-Rojas et al. [3] proposed the voter model (VM) and the contact process model (CP) to simulate the information propagation network and the epidemic spreading network respectively, and studied the interaction between the two models. Ken T. D. Eames [25] modeled a bilayer network containing a social network of parents and an epidemic infection network of children to study the influence of parents' social ties on whether children were vaccinated. In addition, there are some papers about bilayer networks using combinations of these diffusion models: Random Walk [26], Kuramoto [26], Voter [3], Contact [3], LACS [27] and GACS [28]. The above works are all aimed at the bilayer networks, and only one kind of information is spread in each layer. What they are concerned with is not finding the most influential nodes, but the impact of propagation on the whole network from some arbitrary nodes. Unlike them, our work is not limited to the bilayer networks. The number of influences can be arbitrary and they can be freely spread across layers, and what we concern is to find the  $k$  most influential nodes and allocate them fairly.

### Fair seed allocation in a single network and multilayer networks

Whether for a single network or multilayer networks, the previous studies [9, 10, 11, 12, 13, 14] have considered multiple influences, but mostly from the perspective of participants, that is to find the optimal seeds strategy to maximize its own influence on the premise that the opponents have already chosen their seeds. However, participants only have local information of the network, so it is more reasonable to study problems using global information from the perspective of network owners. By selling seeds to its customers, the network platform provides viral marketing services. The network platform not only considers the selection of the most influential seeds, but also considers the equilibrium of seeds allocation to different customers according to their budgets. Fair Seed Allocation (FSA) problem originates from the problem of how to distribute seeds for the network platform which carries out viral marketing. With the in-depth study of the problem of maximizing influence on single network and multilayer networks, it is realized that we should not only consider the maximization of influence, but also consider how to rationally distribute seeds to the customers involved in viral marketing from the perspective of network platform owners in order to maximize the customers' expectations. That is to say, the network platform should not only find the most influential seeds, but also distribute them reasonably to its customers to make them satisfied. FSA is first introduced by Wei Lu et al. [29] under their 'K-LT' model.

Let budgets of customers be  $\gamma_1, \gamma_2, \dots, \gamma_n$ , total budget be  $\gamma = \sum_{i=1}^n \gamma_i$ , and unit price be a constant  $F$ . Fair seed allocation problem is to find a total seed set  $\mathcal{S}$  of minimum size  $k$ , s.t.  $\sigma(\mathcal{S}) \geq F * \gamma$ , and a partition of  $\mathcal{S} = \{S_1, \dots, S_i\}$  to maximize fairness  $f$ :

$$\operatorname{argmax}_{\mathcal{S}} (f = \frac{\min_{i=1}^t \frac{\sigma(S_i, \mathcal{S})}{\gamma_i}}{\max_{i=1}^t \frac{\sigma(S_i, \mathcal{S})}{\gamma_i}}), \quad (2)$$

where  $\sigma(S_i, \mathcal{S})$  is the influence spread of  $S_i$  when  $\mathcal{S}$  is the total seed set.

Ying Yu et al. [30] discussed Fair Seed Allocation problem under their diffusion model 'TIC'. However, they are only studying FSA problem on a single network. Large Internet companies often have more than one online social network, and almost everyone is active in more

than one social network. Finding the most influential seeds in multilayer networks and allocating them to their customers in a balanced way becomes an urgent need to be solved. In this work, we concern FSA problem on multilayer networks. FSA has two phases. The first phase is to find the  $k$  most influential seeds and the second phase is to allocate these seeds fairly to all customers.

## MMIC model and the influence-maximization problem

In this section, we consider the first phase of FSA, i.e., find the  $k$  most influential seeds in multilayer networks. At first, we will propose a diffusion model of multiple influences competing in multilayer networks, called MMIC.

### MMIC model

In a multilayer network, the total seed set is denoted by  $\mathcal{S}$ .  $\{S_1, S_2, \dots, S_t\}$  is a partition of  $\mathcal{S}$ . Elements in  $S_i$  are seeds of influence  $i$ , where  $i = 1, 2, \dots, t$ . We call influence  $i$  **color**  $i$  for convenience, i.e., there are  $t$  colors of seeds. The status of a node is either inactive or active. Initially  $S_i$  is a set of  $i$ -color seeds, in which each node is active with color  $i$ ,  $i = 1, 2, \dots, t$  and all non seeds are inactive. When a node  $u$  is active with color  $i$ , it will attempt to activate its non-seed out-neighbor  $v$  with color  $i$ . Each activation attempts only once with a success probability respectively, and each node can receive  $i$ -activation more than once ( $i = 1, 2, \dots, t$ ), and this inactive-to-active transfer is irreversible. The propagation goes on until there are no more activations. An entity is activated when one of its representatives is active. When the activation process ends, each active entity  $v$  decides to a color  $i$  with probability  $P_v^i$ , where  $P_v^i$  is the proportion of  $i$ -color activations and all activations received by entity  $v$ . In particular, for each entity who has seeds as its representatives decides to color  $i$  with a probability of the proportion of the  $i$ -color seeds among its colored representatives. This diffusion model is called **MMIC**. The **influence spread** of  $S_i$  is the expected number of  $i$ -color entities, denoted by  $\sigma(S_i, \mathcal{S})$ . The **total influence spread** is the expected number of colored entities, denoted by  $\sigma(\mathcal{S})$ .

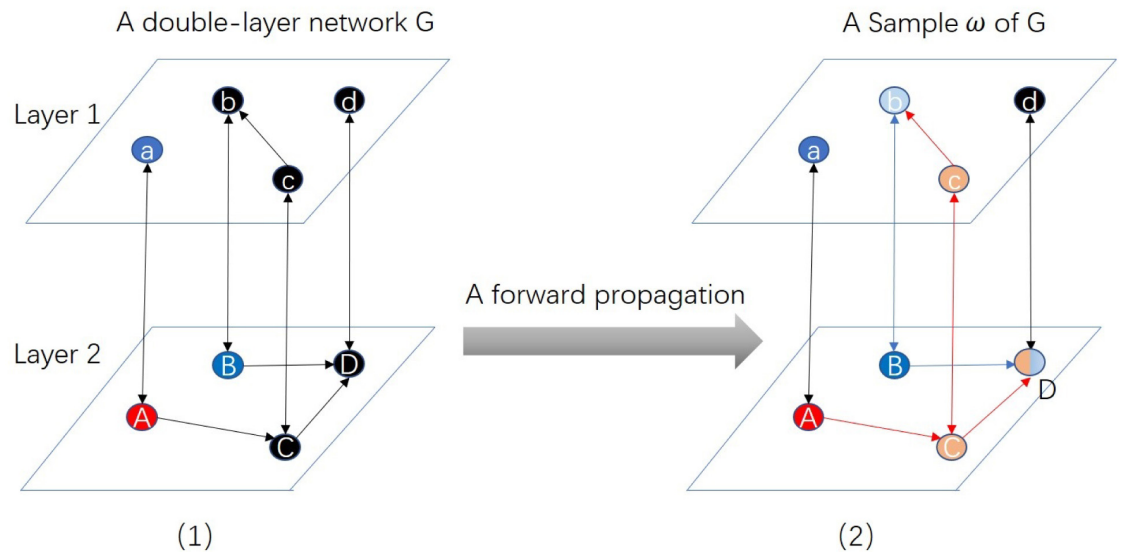
It is easy to verify that  $\sigma(\mathcal{S}) = \sum_{i=1}^t \sigma(S_i, \mathcal{S})$ . Since each active entity has to decide to a color finally, assigning all seeds to a uniform color does not change the total influence spread of  $\mathcal{S}$ .

Fig 2 gives an example of one propagation of multiple influences, i.e., colors in a multilayer network  $G$ . Red and blue nodes are seeds, pink and pale blue nodes are nodes activated by them respectively. Node D receives two types of activation. In the end, the entity 'Dd' can only decide to one color, i.e., red or blue.

Now we study how to calculate the influence spread of each color  $i$ . For each edge  $e$  in network  $G$ , we delete it with probability  $1 - p(e)$ . All of the possible outcomes of this process with their probabilities constitute a probability space  $\Omega$ . Each element of  $\Omega$  is called a **sample** from  $\Omega$ , denoted by  $\omega$ . For a given sample  $\omega$ , a fixed set  $S \subseteq V$  and a given entity  $v_j$ , if there is a path  $\tau$  from  $S$  to one of the representatives of entity  $v_j$ , we call that  $v_j$  is **reachable** from  $S$ , and the path  $\tau$  is called a  **$i$ -color live path** from  $S$  to  $v_j$ . Each edge of path  $\tau$  is called a  **$i$ -color live edge**. For example, 'A-C-D' is a red live path, or we say that entity 'Dd' is reachable from  $S = \{A, C, D\}$  in Fig 2(2).

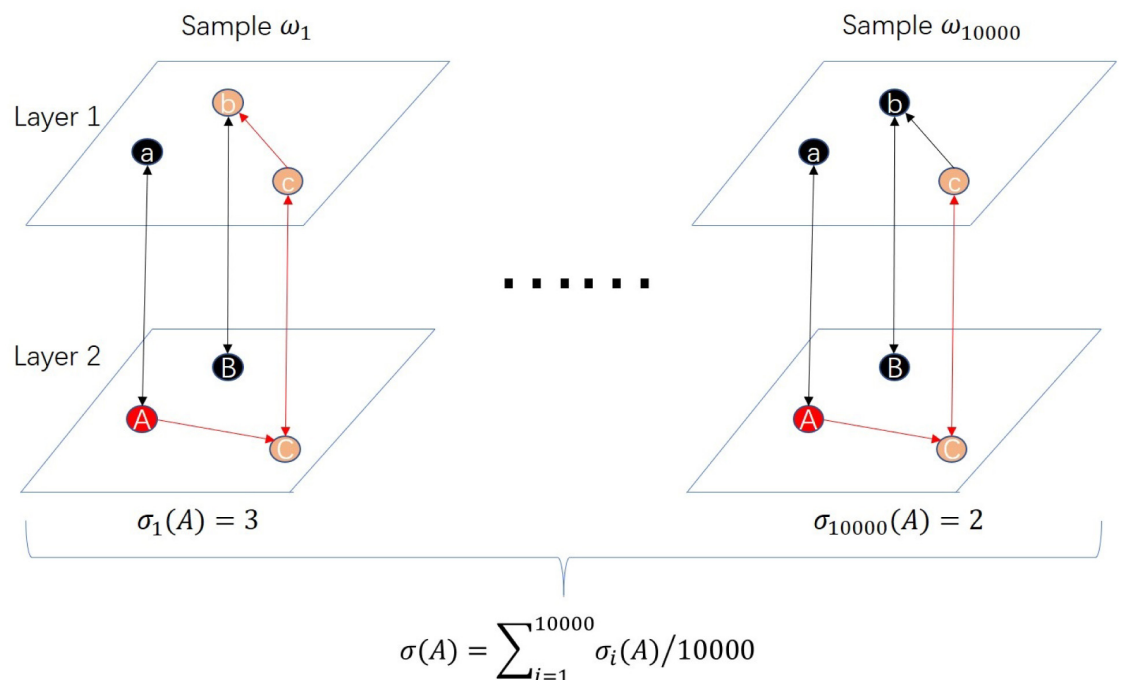
A live path from  $S$  to  $v_j$  indicates that seed set  $S$  can activate entity  $v_j$ . Let  $h_j^\omega(S)$  be the indicator function of that event. In other words,  $h_j^\omega(S) = 1$  when  $v_j$  is reachable from  $S$ , otherwise  $h_j^\omega(S) = 0$ . It can be verified the influence spread of  $S$  is:  $\sigma(S) = \sum \text{Pro}(\omega) \sum_{i=1}^n h_j^\omega(S)$ . However, this formula of influence spread is not practical because the probability of the sample  $\text{Pro}(\omega)$  is difficult to calculate. We use Monte Carlo method to calculate the influence spread  $\sigma(S)$ . Fig 3 gives an example of calculating the influence spread of seed set  $S$ . The red nodes A is the





**Fig 2. An example of one propagation of multiple influences (colors) in a multilayer network  $G$ .** (1) is a double-layer network  $G$ . The uppercase and lowercase of one letter represents one entity, uppercase and lowercase nodes are the representatives of the entity in two layers respectively. Red node 'A', blue node 'a' and blue node 'B' are three seeds of two influences respectively. (2) is an outcome of  $G$  under MMIC, i.e. a sample  $\omega$ . Pink and pale blue nodes are the nodes who have received red and blue activations, respectively. Because entity 'Aa' has one representative 'A' as a red seed, and one representative 'a' as a blue seed, entity 'Aa' decides to each color with a probability of 1/2. Entity 'Bb' only owns blue seed as its representative, so 'Bb' ultimately decides to blue. Entity 'Cc' only receives red activations, so 'Cc' ultimately decides to red. Entity 'Dd' receives red activation once and blue activation once. Therefore, entity 'Dd' decides to each color with a probability of 1/2.

<https://doi.org/10.1371/journal.pone.0229201.g002>



**Fig 3. An example of calculating influence spread using Monte Carlo method.** The 10000 samples are the outcomes of 10000 propagations of  $G$  under MMIC model respectively. The red node is the seed A. The pink nodes are the active nodes in each sample. The dark nodes are the inactive nodes. The red edges are the live edges.  $\sigma_i(A)$  is the number of active entities in sample  $i$ . The influence spread of node A is the average of all  $\sigma_i(A)$  (where  $i = 1, 2, \dots, 10000$ ).

<https://doi.org/10.1371/journal.pone.0229201.g003>

**Table 1. Notations.**

$G(V, E, W)$	multilayer network $G$ with node set $V$ , edge set $E$ , and weight set $W$
$v_j, n$ and $N$	the $j$ th entity of $G$ , the number of entities and the entity set
$\sigma(S)$	the influence spread of $S$
$\Omega$	the probability space of outcomes generated by deleting every edge $e$ with probability $1 - P(e)$
$\omega$	a sample from $\Omega$ , an outcome of one propagation.
$\tau$	a live path
$RRE$	reverse reachable set for entity

<https://doi.org/10.1371/journal.pone.0229201.t001>

seed, i.e.,  $S = \{A\}$ . The pink nodes are the nodes activated by the seed. Each subfigure is the outcome of one propagation. We set the number of iterations to be 10000. The influence spread of the seed is equal to the average number of activated nodes.

We declare that network  $G$  appear in the rest of the paper is multilayer network  $G$ . Many other notations used frequently in this paper are listed in Table 1.

## The influence-maximization problem of MMIC

The first step of FSA problem is seeking  $k$  most influential seeds in  $G$ . So we first consider that all seeds belong to the same color. The influence-maximization problem of IC model is a special case of the problem of MMIC when the weight of every cross-layer edge is 1 or the number of layers is 1. Since the influence-maximization problem of IC model is NP-hard, so is of MMIC. Therefore, there is no polynomial time complexity algorithm to solve this problem. According to [6], if  $\sigma(S)$  is monotone and submodular w.r.t.  $S$ , there is a greedy scheme of seed selection which can provide a  $(1 - 1/e)$ -approximation. Although it can be proved that the influence function  $\sigma(S)$  under MMIC model is monotone and submodular w.r.t.  $S$ , the time complexity of Algorithm 1 (Greedy) is still high, and it is not suitable for large multilayer networks.

### Algorithm 1 Greedy

```

1:  $S' \leftarrow \phi$ 
2: for  $i = 1$  to  $k$  do
3:    $u = \operatorname{argmax}_{u \in V \setminus S'} (\sigma(S' \cup \{u\}) - \sigma(S'))$ ;
4:   if  $\sigma(S' \cup \{u\}) - \sigma(S') > 0$  then
5:      $S' \leftarrow S' \cup \{u\}$ 
6:   end if
7: end for
8: return  $S'$ 

```

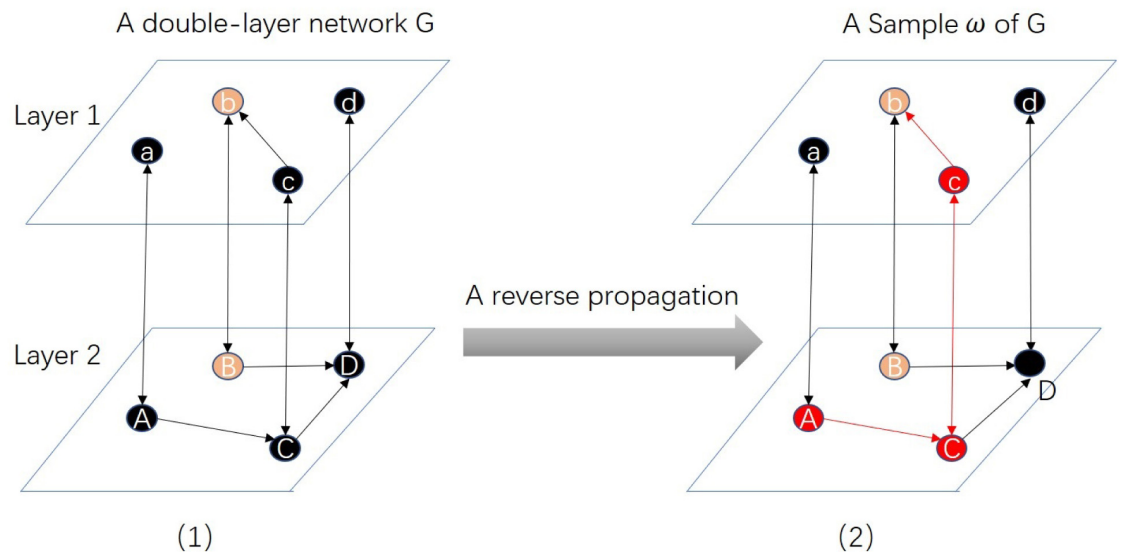
Intuitively, nodes with large out-degree are generally influential. Based on this idea, we use the following method to select seeds. First we select the node with largest out-degree, and then delete it from the network. After repeating this process  $k$  times we have  $k$  seeds, see Algorithm 2.

### Algorithm 2 Degree

```

1:  $S'' \leftarrow \phi$ 
2:  $G' = G$ 
3: for  $i = 1$  to  $k$  do
4:    $u = \operatorname{argmax}_{u \in G'} \operatorname{outdegree}(u)$ ;
5:    $S'' \leftarrow S'' \cup \{u\}$ 
6:    $G' = G' - u$ 
7: end for
8: return  $S''$ 

```



**Fig 4. An example of random RRE of entity  $Bb$  in a multilayer network.** (1) is a bilayer network  $G$ , the pink nodes ' $B$ ' and ' $b$ ' are representatives of entity  $Bb$ . After a reverse propagation (along the opposite direction of edges) from ' $B$ ' and ' $b$ ', we have a sample  $\omega$ , i.e., (2). The active nodes ( $B, b, c, C, A$ ) which constitute a random RRE of entity  $Bb$ .

<https://doi.org/10.1371/journal.pone.0229201.g004>

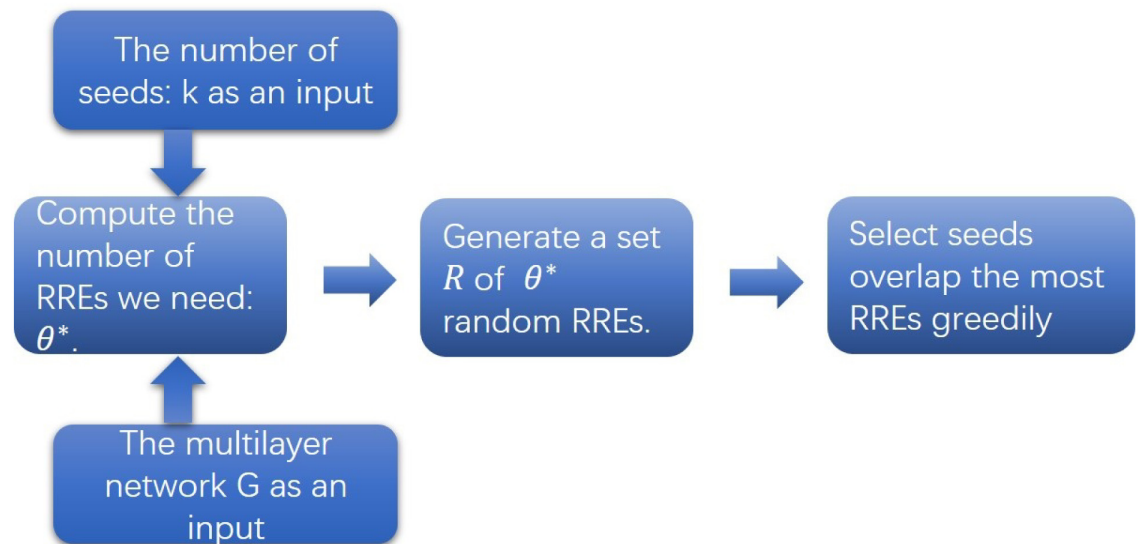
The advantage of the degree method is that the computing speed is very fast, but there is no theoretical guarantee of its performance, i.e., its performance varies with different networks.

Youze Tang et al. [16] proposed an algorithm IMM to solve the influence-maximization problem on a single network. It is the state-of-art algorithm we have known which has both guarantee of approximation and nearly linear time complexity. The core idea of IMM is based on the concept of reverse reachable (RR) set. A RR set of node  $u$  consists of all nodes from which  $u$  is reachable in sample  $\omega$ , i.e., it contains all nodes can directly or indirectly activate node  $u$  in sample  $\omega$ . Furthermore, if node  $u$  is chosen uniformly at random from  $V$ , a RR set is called a random RR set. Suppose we have  $\theta$  (large enough, for example,  $\theta = 10000$ ) random RR sets from  $\theta$  random nodes. If a node  $x$  appears 9999 times in these 10000 RR sets, we say that node  $x$  overlaps 9999 RR sets. From the definition of RR set, we can infer that  $x$  has great influence spread because it has the ability to activate many of these 10000 nodes as a seed. For a given node set  $S$ , that  $S$  overlaps more random RR sets indicates that  $S$  is capable of activating more nodes as the seed set. Therefore the influence-maximization problem of IC model is transferred to seeking for a node set  $S$  of size  $k$  to overlap the most random RR sets. For our MMIC model, the influence spread is the expected number of final active entities instead of nodes, so we need to define reverse reachable set for entity.

Let  $v_j$  be an entity of multilayer network  $G$ , and  $\omega$  be a sample from  $\Omega$ . Let  $R_j^\omega$  be the set consists of all nodes from which  $v_j$  can be reachable. The set  $R_j^\omega$  is called the **reverse reachable set of entity  $v_j$ , RRE** for short. If  $v_j$  is chosen uniformly at random from  $N$ , and  $\omega$  is a random outcome of MMIC, the RRE is called a random RRE.

Fig 4 shows an example of random RRE of entity ' $Bb$ ' in a multilayer network. After a random reverse propagation (along the opposite direction of the edges) from  $B$  and  $b$  (the pink nodes), we have the outcome of active nodes (the red and pink nodes):  $R_1 = \{A, B, C, b, c\}$  which is a random RRE of entity ' $Bb$ '.





**Fig 5. The procedure of RRE method (Algorithm 3).**

<https://doi.org/10.1371/journal.pone.0229201.g005>

Fig 5 shows the procedure of RRE method. Firstly, we calculate the number of random RREs we needed:  $\theta^*$  (The value of  $\theta^*$  is calculated by algorithm IMM from [16]). Secondly, we generate  $\theta^*$  random RREs. At last, we add nodes that overlap the most random RREs as seeds greedily. RRE method of influence-maximization problem under MMIC model is presented as Algorithm 3.

**Algorithm 3** RRE ( $\theta^*$ , R)

```

1: Compute  $\theta^*$  and R from algorithm 2 in [16]
2: while  $|R| \leq \theta^*$  do
3:   Select an entity  $v_j$  from G uniformly at random;
4:   Generate a random RRE of  $v_j$  and insert it into R;
5: end while
6:  $S^* \leftarrow \phi$ 
7: for  $i = 1$  to  $k$  do
8:    $u = \operatorname{argmax}_u (F_R(S^* \cup \{u\}) - F_R(S^*))$ ;
9:    $S^* \leftarrow S^* \cup \{u\}$ 
10: end for
11: return  $S^*$ 
  
```

Our Algorithm 3 extends the RR set of the algorithm 2 in [16] to RRE, which does not change the time complexity, so according to [16], the time complexity of our Algorithm 3 is still  $O\left(\frac{(k+l)(n+m)\log n}{\epsilon^2}\right)$ , where  $n$  and  $m$  are the numbers of nodes and edges, respectively in the multilayer networks,  $k$  is the number of seeds, The two parameters  $l$  and  $\epsilon$  are set to the same value as in [16]. Next, we will demonstrate the effectiveness and robustness of RRE method relative to degree method through simulation.

**Simulation.** *Experimental setup.* We use ‘Stanford Large Network Dataset Collection’ from [31]. Two real social networks are selected: Wikipedia vote network and Bitcoin Alpha trust weighted signed network. Wikipedia is a free encyclopedia written by volunteers from all over the world. Anyone who wants to participate in managing the site needs to submit an application to the Wikipedia community for public discussion and voting. Wiki-Vote network

Table 2. Network info.

Network	#Nodes	#Edges	Average in-degree	Average out-degree
Wiki-Vote	7,115	103,689	14.5733	14.5733
Bitcoin-alpha	3,783	24,186	6.3933	6.3933
Wiki-Vote0	14,172	200,814	14.1698	14.1698
Bitcoin-alpha0	7,564	51100	6.7557	6.7557

<https://doi.org/10.1371/journal.pone.0229201.t002>

contains all the voting data of Wikipedia from the inception of Wikipedia to January 2008. The nodes in the network Wiki-Vote represent Wikipedia users, and the directed edges from node  $i$  to node  $j$  indicate that user  $i$  voted for user  $j$ . Bitcoin Alpha is a Bitcoin trading platform. Since Bitcoin transactions are anonymous, users need to maintain their reputation to prevent transactions with users at risk of fraud. Each user grades the others from -10 (total distrust) to +10 (total trust). The nodes represent the users, the edges represent the trust relationships between them, which constitute the network Bitcoin-alpha. By randomly deleting ten percent of the edges of Wiki-Vote respectively, we get two networks named Wiki-Vote1 and Wiki-Vote2. After adding cross-layer edges between them, we get multilayer network Wiki-Vote0. Network Bitcoin-alpha0 is also constructed in the same way. The way to assign weights is WC. That is, the weight of edge  $(u, v)$  is the reciprocal of  $v$ 's in-degree. WC is one of the most commonly used weighting methods because it is a reasonable normalized measure. That is to say, when we do not know the probabilities of node  $u$  activating its out-neighbors, a reasonable way to define these probabilities is to assume the probabilities are all the same, and the sum of them is 1. The basic statistics of the datasets are shown in Table 2.

Here is our lab server configuration. Operating system: CentOS; CPU: Intel (R) Xeon (R) E5, 32 core; Memory 64G.

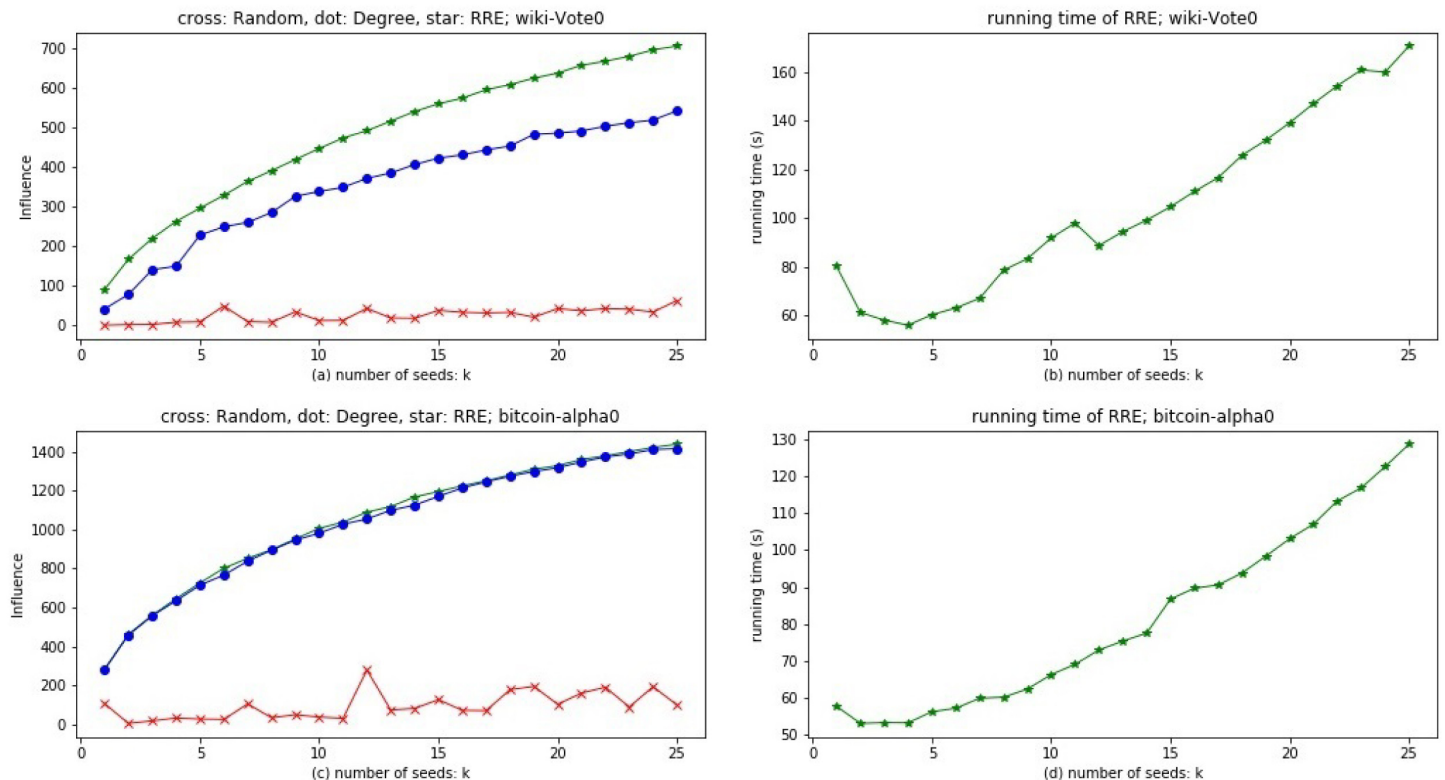
**Evaluation method.** Our evaluation index is the influence spread of selected  $k$  seeds by different methods. We do experiments on network Wiki-Vote0 and Bitcoin-alpha0 with three methods: random, degree and RRE. Because the time complexity of greedy algorithm is too high, we do not use it in experiments.

The experimental results of influence maximization are shown in Fig 6. Nodes with large out degrees often have great influence spread, so the degree method can sometimes achieve good results, such as the results on Bitcoin-alpha0, see Fig 6(c). But for some multilayer networks, such as the results on Wiki-Vote0, see Fig 6(a), the degree method is not very effective. The reason is that it ignores nodes with small degree but associated with important nodes. RRE method is more robust than degree method because it can reach nearly  $(1 - 1/e)$  of the optimal influence spread in the worst case for any multilayer networks. From Fig 6, we can see that RRE method outperforms others on both Wiki-Vote0 and Bitcoin-alpha0. Considering time complexity and robustness, we use RRE method to select the  $k$  most influential seeds for FSA problem.

## Fair seed allocation problem of MMIC model

After finding the  $k$  most influential seeds by RRE method, we consider the issue of fairly allocating the  $k$  seeds to  $t$  different customers (colors) from the perspective of the platform owner, which is the second phase of FSA. That is to say, make sure that for each color, the expected influence spread is proportional to its budget.

Now we allocate  $k$  seeds to  $t$  different colors to maximize  $f$  in Eq (2). Before that, some preparatory knowledge should be given first.



**Fig 6. The influence spread of three methods and the running time of RRE method.** The number of iterations to calculate influence spread is 10000. For (a) and (c), abscissa is the number of seeds; ordinates represent the influence spread; red cross, blue dot and green star line represent the methods of random, degree and RRE respectively. For (b) and (d), abscissa is the number of seeds, ordinates represent the running time (s) of RRE method.

<https://doi.org/10.1371/journal.pone.0229201.g006>

Given a sample  $\omega$ , let  $a^\omega(S_i, v)$  be the number of  $i$ -color live paths from  $i$ -color seed set  $S_i$  to entity  $v$ . Let  $b^\omega(S_i, \mathcal{S}, v)$  be the probability of that entity  $v$  is  $i$ -colored as  $S_i$  is the  $i$ -color seed set and  $\mathcal{S}$  is the total seed set. Let  $c_{S_i}$  be the number of representatives of  $v$  in  $S_i$  and  $c$  be the number of representatives of  $v$  in  $\mathcal{S}$ . Let  $N_1$  be the set of entities who has at least one seed as their representative. Then we have:

$$b^\omega(S_i, \mathcal{S}, v) = \begin{cases} \frac{a^\omega(S_i, v)}{a^\omega(\mathcal{S}, v)}, & \text{if } v \notin N_1. \\ \frac{c_{S_i}}{c}, & \text{if } v \in N_1. \end{cases} \quad (3)$$

If  $a^\omega(\mathcal{S}, v) = 0$ , i.e., there is no live path from  $\mathcal{S}$  to  $v$ , we specify that  $b^\omega(S_i, \mathcal{S}, v) = 0$ .

**Theorem 1.**

$$\sigma(S_i, \mathcal{S}) = \sum_{s \in S_i} \sigma(s, \mathcal{S}) \quad (4)$$

*Proof.*

$$\begin{aligned}
 \sigma(S_i, \mathcal{S}) &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \sigma^\omega(S_i, \mathcal{S}) \\
 &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N} b^\omega(S_i, \mathcal{S}, v) \\
 &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \left[ \sum_{v \in N \setminus N_1} \frac{a^\omega(S_i, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{c_{S_i}}{c} \right] \\
 &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N \setminus N_1} \frac{a^\omega(S_i, v)}{a^\omega(\mathcal{S}, v)} + \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N_1} \frac{c_{S_i}}{c} \\
 &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N \setminus N_1} \frac{a^\omega(S_i, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{c_{S_i}}{c} \\
 &= \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N \setminus N_1} \sum_{s \in S_i} \frac{a^\omega(s, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{\sum_{s \in S_i} c_s}{c} \\
 &= \sum_{s \in S_i} \left[ \sum_{\omega \in \Omega} \text{Pro}(\omega) \sum_{v \in N \setminus N_1} \frac{a^\omega(s, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{c_s}{c} \right] \\
 &= \sum_{s \in S_i} \sum_{\omega \in \Omega} \text{Pro}(\omega) \left[ \sum_{v \in N \setminus N_1} \frac{a^\omega(s, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{c_s}{c} \right] \\
 &= \sum_{s \in S_i} \sum_{\omega \in \Omega} \text{Pro}(\omega) \sigma^\omega(s, \mathcal{S}) \\
 &= \sum_{s \in S_i} \sigma(s, \mathcal{S})
 \end{aligned}$$

Recall that  $\sigma(\mathcal{S}) = \sum_{i=1}^t \sigma(S_i, \mathcal{S})$ , then we have  $\sigma(\mathcal{S}) = \sum_{i=1}^t \sum_{s \in S_i} \sigma(s, \mathcal{S})$ . That means the influence spread of  $S_i$  is the sum of influence spread of every element  $s$  in  $S_i$  while  $s$  is the only  $i$ -seed in  $\mathcal{S}$ , and the total influence spread is the sum of influence spread of every element  $s$  in  $\mathcal{S}$  while  $s$  is the only  $i$ -seed in  $\mathcal{S}$ .

**Theorem 2.**  $\sigma(S_i, \mathcal{S})$  is monotone and submodular w.r.t.  $S_i$ .

*Proof.* By the proof of Theorem 1,

$$\sigma(S_i, \mathcal{S}) = \sum_{\omega \in \Omega} \text{Pro}(\omega) \left[ \sum_{v \in N \setminus N_1} \frac{a^\omega(S_i, v)}{a^\omega(\mathcal{S}, v)} + \sum_{v \in N_1} \frac{c_{S_i}}{c} \right].$$

Since  $\text{Pro}(\omega)$ ,  $a^\omega(\mathcal{S}, v)$  and  $c$  are all non-negative and independent of  $S_i$ , we have that  $\sigma(S_i, \mathcal{S})$  is a non-negative linear combination of  $a^\omega(S_i, v)$  and  $c_{S_i}$ . All we need to do is to prove that  $a^\omega(S_i, v)$  and  $c_{S_i}$  are both monotone and submodular w.r.t.  $S_i$ .

(i) Monotonicity

Given a sample  $\omega$ , for arbitrary  $x \in V$ , the number of  $i$ -color live paths from  $i$ -color seed set  $S_i \cup \{x\}$  to entity  $v$  is  $a^\omega(S_i \cup \{x\}, v) = a^\omega(S_i, v) + a^\omega(x, v)$ . Therefore  $a^\omega(S_i \cup \{x\}, v) - a^\omega(S_i, v) = a^\omega(x, v) \geq 0$ .  $a^\omega(S_i, v)$  is monotone w.r.t.  $S_i$ .

Similarly,  $c_{S_i \cup \{x\}} - c_{S_i} = c_x \geq 0$ , then  $c_{S_i}$  is monotone w.r.t.  $S_i$ .

(ii) Submodularity

For arbitrary  $x \in V$  and  $S_i \subseteq S'_i \subseteq V$ , there are two cases:

(Case 1) If  $x \in S'_i$ ,  $a^\omega(S'_i \cup \{x\}, v) - a^\omega(S'_i, v) = 0$ . Then we have

$$a^\omega(S_i \cup \{x\}, v) - a^\omega(S_i, v) \geq a^\omega(S'_i \cup \{x\}, v) - a^\omega(S'_i, v) = 0.$$

(Case 2) If  $x \notin S'_i$ , we have  $a^\omega(S'_i \cup \{x\}, v) - a^\omega(S'_i, v) = a^\omega(x, v) = a^\omega(S_i \cup \{x\}, v) - a^\omega(S_i, v)$ .

Therefore,  $a^\omega(S_i, v)$  is submodular w.r.t.  $S_i$ .

Similarly, it is easy to verify that  $c_{S_i}$  is submodular w.r.t.  $S_i$ .

In summary,  $\sigma(S_i, \mathcal{S})$  is monotone and submodular w.r.t.  $S_i$ .

By Theorem 1 and 2, the seed allocation phase of FSA problem is demonstrated as follows:

Firstly, compute all  $\sigma(s, \mathcal{S})$  for any  $s \in \mathcal{S}, j = 1, 2, \dots, k$ . Note that  $\mathcal{S}$  is the set of  $k$  seeds from Algorithm 3. Secondly, allocate all  $k$  elements in  $\mathcal{S}$  to  $t$  seed sets  $S_1, S_2, \dots, S_t$  to make the differences among  $f_i = \frac{\sum_{s \in S_i} \sigma(s, \mathcal{S})}{\gamma_i}, i = 1, 2, \dots, t$  minimized. In other words, maximizing  $f = \frac{\min_{i=1}^t f_i}{\max_{i=1}^t f_i}$ , where  $f_i$  and  $f$  are defined in Eq (2). We are going to propose two greedy methods for seed allocation which are guaranteed by the following theorem:

**Theorem 3.** Assume that  $\mathcal{S}$  is the total seed set obtained in the first phase of FSA problem. Allocating  $s \in \mathcal{S}$  one by one to  $i$ -color seed set  $S_i$  who has the current least  $f_i$  is the greedy choice of maximizing  $f$ .

*Proof.* Now we have to prove that allocating seed  $s$  to  $i$ -color who has the current least  $f_i$  is the greedy choice of maximizing  $f$ . In other words, it is locally optimal in each step of allocation. We use reduction to absurdity to prove that.

Without losing generality, assume that  $f_1 \leq f_2 \leq \dots \leq f_t$  currently, but we allocate seed  $s$  to  $j$ -color,  $j \neq 1$ . Let  $f'_j$  be the updated value of  $f_j$  when the allocation has been done, while the other  $f_i, i \neq j$  remain the same according to Theorem 1.

(Case 1) If  $f'_j \leq f_i$ , the minimum and maximum of  $f_i, i = 1, 2, \dots, t$  is not changed. Therefore, we have the updated value of  $f$  as  $f' = \frac{f_1}{f_t}$ . However, the updated value of  $f$  would be  $f'' = \frac{\min\{f'_1, f'_2\}}{f_t}$  if  $s$  is allocated to color 1. By Theorem 1 and 2,  $f' < f''$ , then allocating  $s$  to color  $j$  is not the greedy choice.

(Case 2) If  $f'_j > f_i$ , we have the updated value of  $f$  as  $f' = \frac{f_2}{f'_j}$ . If  $s$  is allocated to color 1, there are two subcases.

(Subcase 1) If  $f'_1 \leq f_i$ , we have  $f'' = \frac{\min\{f'_1, f'_2\}}{f_t}$  as the updated value of  $f$  if  $s$  is allocated to color 1. By Theorem 1 and 2,  $f_1 \leq \min\{f'_1, f'_2\}$  and  $f'_j > f_i$ , then we have  $f' < f''$ .

(Subcase 2) If  $f'_1 > f_i$ , we have  $f'' = \frac{f_2}{f'_1}$  as the updated value of  $f$  if  $s$  is allocated to color 1. By Theorem 1 and 2,  $f_1 \leq f_2$  and  $f'_j > f'_1$ , then we have  $f' < f''$ .

In both subcases, allocating  $s$  to color  $j, j \neq 1$  is not the greedy choice. In summary, allocating  $s$  to color  $i$  who has the current least  $f_i$  is the greedy choice of maximizing  $f$ .

The orders of seed allocation are specified in a way from [30]. Firstly, we get  $k$  and  $\mathcal{S}$  as the output of Algorithm 3. Secondly, we initialize all seed sets  $S_i$  to empty sets and initialize all their  $f_i$  to 0. At last, we allocate  $s \in \mathcal{S}$  one by one by its  $\sigma(s, \mathcal{S})$  non-decreasingly into  $S_i$  with the current least  $f_i$  (by Theorem 3). In accordance with this idea, we propose Algorithm 4 to allocate seeds.

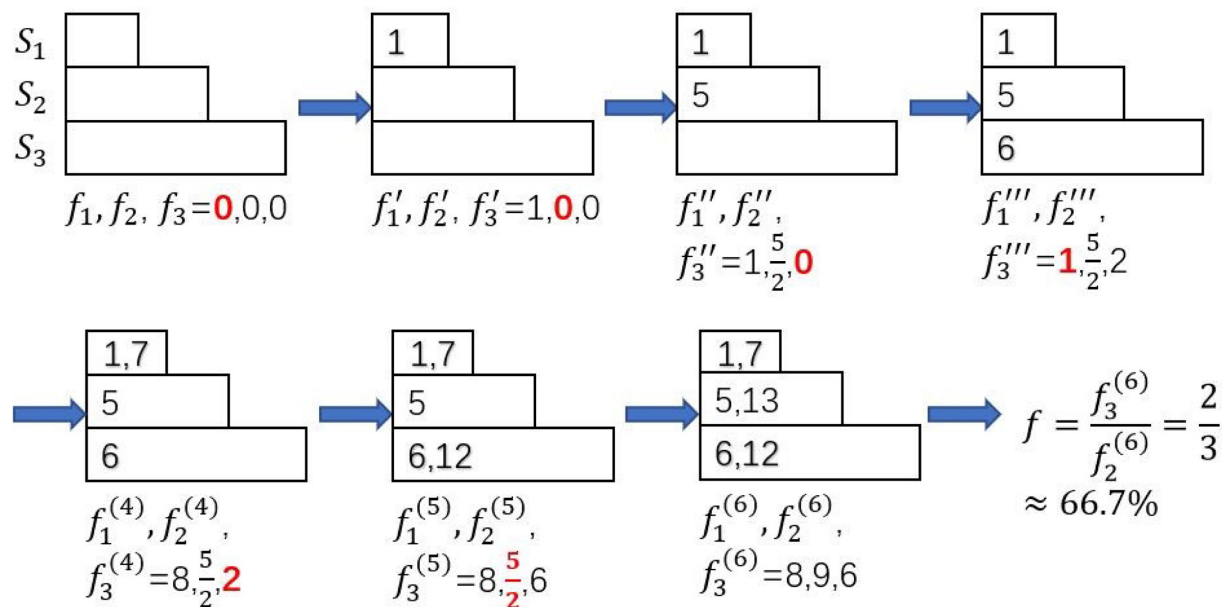
#### Algorithm 4 Fair1

```

1:  $k = \min\{k = |\mathcal{S}^*| : \sigma(\mathcal{S}^*) - F * \gamma \geq 0\}$ , where  $\mathcal{S}^*$  is the output of Algorithm 3.
2:  $\mathcal{S} \leftarrow \mathcal{S}^*$ ;
3:  $S_i \leftarrow \emptyset, i = 1, 2, \dots, t$ ;
4:  $f_i \leftarrow 0, i = 1, 2, \dots, t$ ;
5: Sort the elements of  $\mathcal{S}$  into  $\{s_1, s_2, \dots, s_k\}$  by their  $\sigma(s_j, \mathcal{S})$  non-decreasingly;
6: for  $j = 1$  to  $k$  do
7:    $S_{i^*} \leftarrow S_{i^*} \cup \{s_j\}$ , where  $i^* = \arg_{1 \leq i \leq t} \min f_i$ ;
8:    $f_{i^*} = \frac{\sigma(S_{i^*}, \mathcal{S})}{\gamma_{i^*}}$ ;
9: end for
10: return  $\{S_1, S_2, \dots, S_t\}$ 

```

Here is an example of how Algorithm 4 and 5 work. We assume there are three colors with budgets  $\gamma_1: \gamma_2: \gamma_3 = 1: 2: 3$ , and the number of seeds  $k = 6$ . Suppose that after the first phase of



**Fig 7. The procedure of Fair1 (Algorithm 4) when budgets are 1:2:3 and  $\sigma(s_i, S) = 1, 5, 6, 7, 12, 13$  for  $i = 1, 2, \dots, 6$ .**

<https://doi.org/10.1371/journal.pone.0229201.g007>

FSA, we obtain six seeds:  $s_j, j = 1, 2, \dots, 6$ , with their influence spread  $\sigma(s_j, S) = 1, 5, 6, 7, 12, 13$  for  $j = 1, 2, \dots, 6$  calculated by Algorithm 3.

Now, we use Algorithm 4 for seed allocation phase. The size of the three rectangles in Fig 7 represents the size of the budgets. In this phase, we should allocate the six seeds to three colors ( $S_1, S_2, S_3$ ) to maximize  $f$ . We initialize all seed sets  $S_i$  to empty sets and set all initial values of  $f_i$  to be 0. We allocate  $s_j$  one by one (by their  $\sigma(s_j, S)$  non-decreasingly) into  $S_i$  who has the least value of  $f_i$  (in red) currently, and update the value of  $f_i$ . The final allocation of seeds is  $S_1 = \{s_1, s_4\}, S_2 = \{s_2, s_6\}, S_3 = \{s_3, s_5\}$  and  $f \approx 66.7\%$ .

Now we will propose another method for the orders of allocations. Firstly we get  $k$  and  $S$  as the output of Algorithm 3. Secondly, we initialize all seed sets  $S_i$  to empty sets, and initialize  $f_i$  to the reciprocals of its budget. At last, we allocate  $s \in S$  one by one by its  $\sigma(s, S)$  non-increasingly into  $S_i$  with the current least  $f_i$  (by Theorem 3). Therefore, we obtain another method of seed allocation, i.e., Algorithm 5:

#### Algorithm 5 Fair2

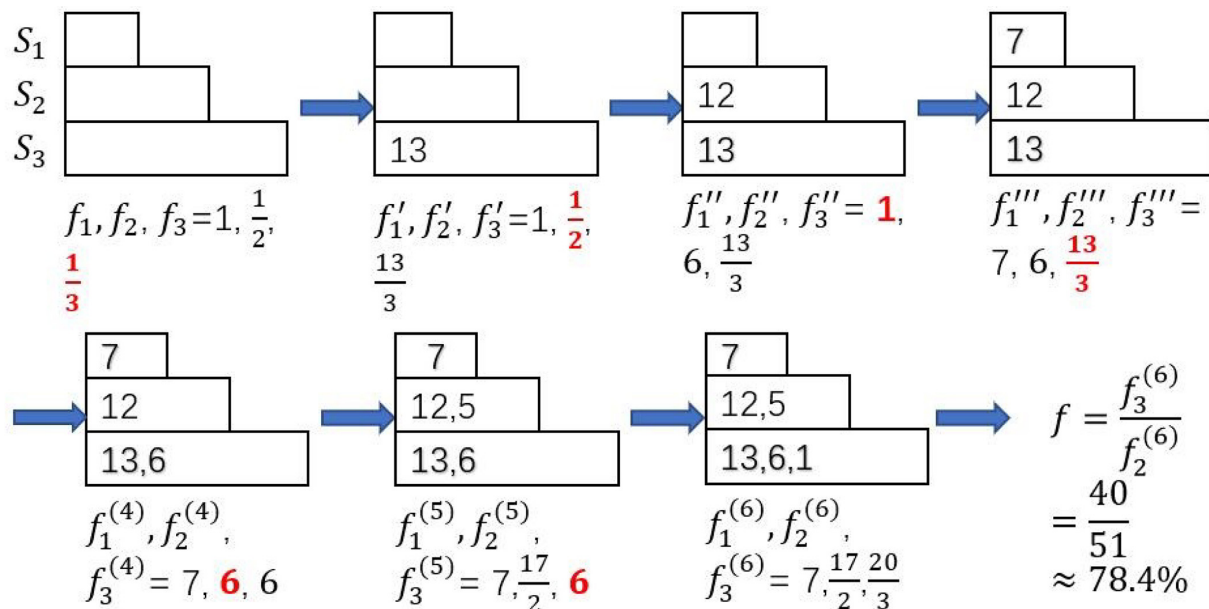
```

1:  $k = \min\{k = |S^*| : \sigma(S^*) - F^* \cdot \gamma \geq 0\}$ , where  $S^*$  is the output of Algorithm 3.
2:  $S \leftarrow S^*$ ;
3:  $S_i \leftarrow \emptyset, i = 1, 2, \dots, t$ ;
4:  $f_i \leftarrow \frac{1}{\gamma_i}, i = 1, 2, \dots, t$ ;
5: Sort the elements of  $S$  into  $\{s_1, s_2, \dots, s_k\}$  by their  $\sigma(s_j, S)$  non-increasingly;
6: for  $j = 1$  to  $k$  do
7:    $S_{i^*} \leftarrow S_{i^*} \cup \{s_j\}$ , where  $i^* = \arg\min_{1 \leq i \leq t} f_i$ ;
8:    $f_{i^*} = \frac{\sigma(s_j, S)}{\gamma_{i^*}}$ ;
9: end for
10: return  $\{S_1, S_2, \dots, S_t\}$ 

```

Now, we use Algorithm 5 for seed allocation phase in stead of Algorithm 4. The process is shown in Fig 8. We set initial value of  $f_i$  to be the reciprocal of its budget, that is  $\frac{1}{\gamma_i}$ . We allocate





**Fig 8.** The procedure of Fair2 (Algorithm 5) when budgets are 1:2:3 and  $\sigma(s_i, S) = 1, 5, 6, 7, 12, 13$  for  $i = 1, 2, \dots, 6$ .

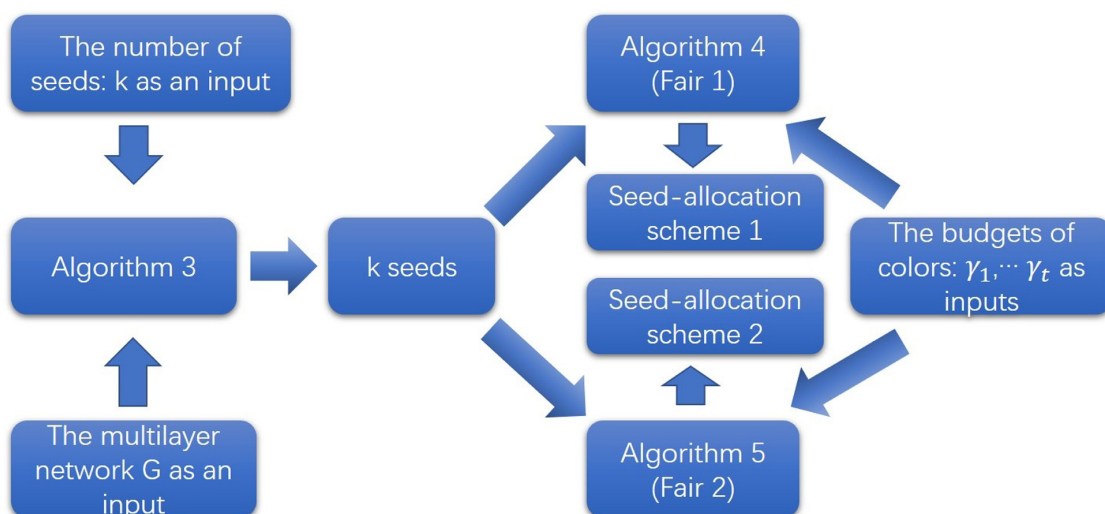
<https://doi.org/10.1371/journal.pone.0229201.g008>

$s_j$  one by one (by their  $\sigma(s_j, S)$  non-increasingly) into  $S_i$  who has the least value of  $f_i$  (in red) currently, and update the value of  $f_i$  meanwhile until all six seeds are allocated. The final allocation of seeds is  $S_1 = \{s_4\}$ ,  $S_2 = \{s_2, s_5\}$ ,  $S_3 = \{s_1, s_3, s_6\}$  and  $f \approx 78.4\%$ .

In summary, Fig 9 is a flowchart illustrating our solutions of FSA problem.

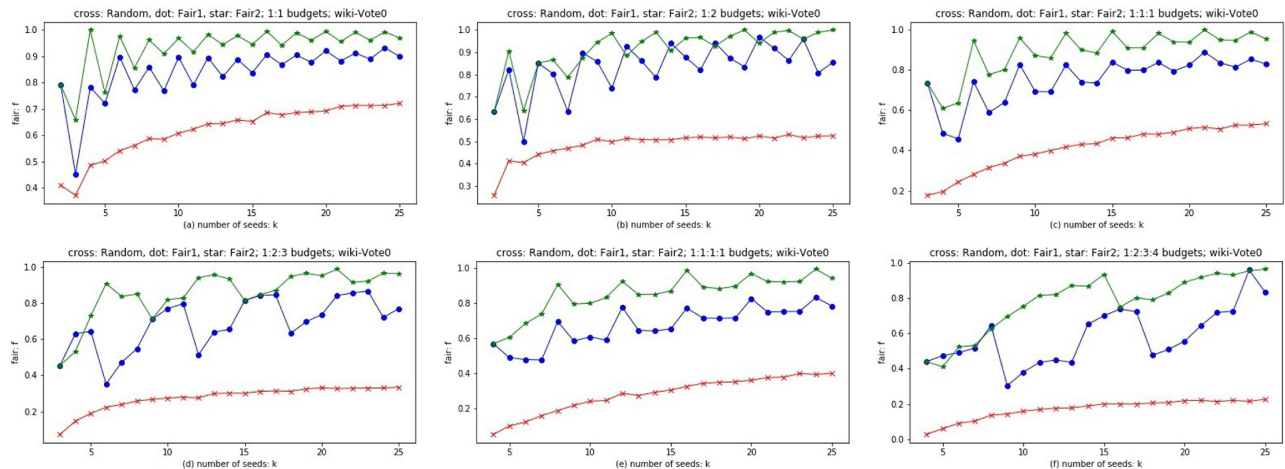
## Simulation

Considering that there are different budget configurations in real world, we choose both balanced budgets of 1:1, 1:1:1, 1:1:1:1, and unbalanced budgets of 1:2, 1:2:3, 1:2:3:4. Experiments are conducted on Wiki-Vote0 and Bitcoin-alpha0 respectively. The abscissa is the number of



**Fig 9.** The procedure of solutions of Fair seed allocation problem.

<https://doi.org/10.1371/journal.pone.0229201.g009>



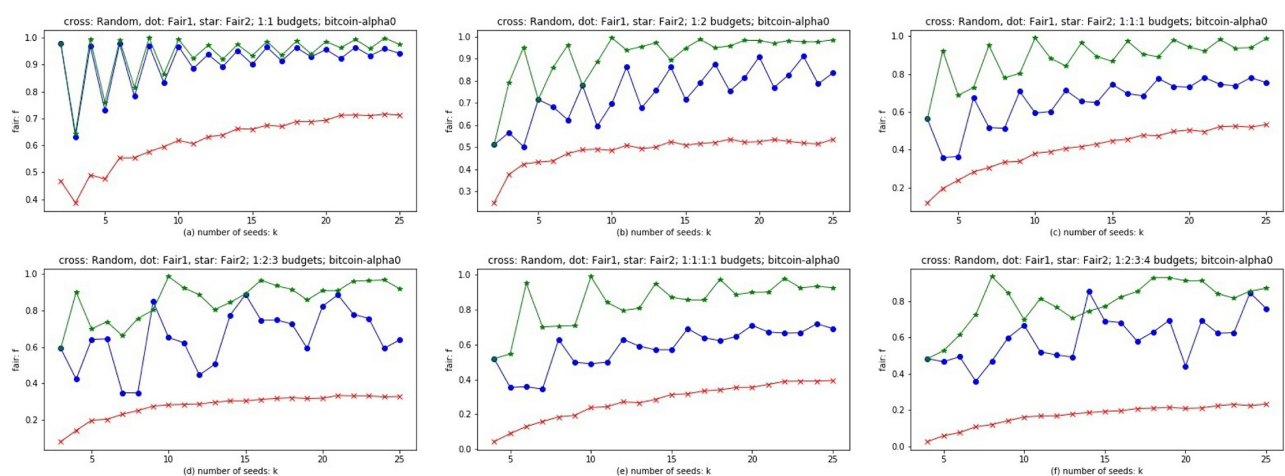
**Fig 10.** The results of Fair1 and Fair2 on multilayer network Wiki-Vote0. Abscissa is the number of seeds; ordinates represent the value of  $f$ ; red cross, blue dot and green star line represent Random, Fair1 and Fair2 respectively. Subfigure (a)-(f) are of different proportion of budgets.

<https://doi.org/10.1371/journal.pone.0229201.g010>

seeds:  $k$ , the ordinate is fairness  $f$ . Three methods are considered, Random (red cross line), Fair1 (blue dot line), and Fair2 (green star line). The result of Random method is the average result of 10,000 Monte Carlo simulations. The closer  $f$  is to 1, the more effective the method is. The time complexity of Algorithm 4 and 5 of seed allocation phase is  $O(k^*t)$ , where  $k$  is the number of seeds and  $t$  is the number of colors (customers).

## Discussion

From the experimental results shown in Figs 10 and 11, Fair1 and Fair2 significantly outperform random seed allocation, and Fair2 has the best results on different budgets for different networks. According to Theorem 3, both Fair1 and Fair2 are greedy strategies to maximize fairness  $f$ . The curves of Fair1 and Fair2 are oscillating, but with the increase of the number of seeds  $k$ , the amplitude of the oscillation decreases. No matter for Fair1 or Fair2, which color



**Fig 11.** The results of Fair1 and Fair2 on multilayer network Bitcoin-alpha0 abscissa is the number of seeds; ordinates represent the value of  $f$ ; red cross, blue dot and green star line represent Random, Fair1 and Fair2 respectively. Subfigure (a)-(f) are of different proportion of budgets.

<https://doi.org/10.1371/journal.pone.0229201.g011>

the next seed is assigned to is based on the current optimal choice, i.e., allocating the coming seed to color  $i$  with the minimum  $f_i$  (If more than one  $f_i$  is equal to the minimum, choose a color  $i$  from these  $f_i$  randomly). If the current  $f$  is close to 0, i.e., the gap between the minimum and the maximum of  $\{f_1, f_2, \dots, f_t\}$  is very large, then allocating a new coming seed usually increase the value of  $f$ . On the contrary, if the current  $f$  is close to 1, allocating a new coming seed usually break the balance among  $\{f_1, f_2, \dots, f_t\}$  and even make  $f$  smaller. This is the potential reason for the oscillations. With the increase of  $k$ , i.e., the number of allocated seeds, the ratio of a new coming seed's influence spread to allocated seeds' influence spread is usually getting smaller, especially for Fair2, because the influence spread of the newly allocated seed is non-increasing. Therefore, allocating a new coming seed has less effect on  $f$  with the increase of  $k$ . This is the potential reason of the amplitude decrement and the convergence of curves with the increase of  $k$ . The difference between Fair1 and Fair2 is that Fair1 allocates seeds by their influence spread from small to large, while Fair2 allocates seeds by their influence spread from large to small. Recall that, the same allocation strategy of Fair1 and Fair2 is allocating the coming seed to color  $i$  with the minimum  $f_i$  (If more than one  $f_i$  is equal to the minimum, choose a color  $i$  from these  $f_i$  randomly). For Fair1, the allocations of the first  $t$  seeds are all random because the initial values of  $f_i$  ( $i = 1, 2, \dots, t$ ) are all equal to zeros. For Fair2, since the initial values of  $f_i$  ( $i = 1, 2, \dots, t$ ) are the reciprocals of the budgets respectively, they may be different from each other, i.e., the allocations of the first  $t$  seeds may not be random. Therefore, Fair2 is more reasonable than Fair1 for the first  $t$  allocations. On the other hand, as we discussed before, with the increase of  $k$ , the oscillation of Fair2 decreases faster than that of Fair1. Therefore, Fair2 is superior to Fair1 in almost all cases.

## Conclusion

In this work, we propose a multi-influence diffusion model MMIC for multilayer social networks. Unlike traditional models of single influence in a single network, MMIC model considers not only multilayer networks, but also multiple competing influences propagating within them. From the point of view of the network owner, we propose a Fair Seed Allocation problem for multilayer networks. Firstly, we propose 'RRE method' as Algorithm 3 to find the  $k$  most influential seeds. Then we allocate the  $k$  seeds to  $t$  different colors (customers) according to Fair1 (Algorithm 4) or Fair2 (Algorithm 5) so that their influence spread is proportional to their budgets. We proved theoretically that the two allocation strategies are greedy choices. Our experiments on real social networks show the effectiveness of our methods.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions. We thank Rongyan Cai for her kind help.

## Author Contributions

**Conceptualization:** Yu Chen, Wei Wang.

**Data curation:** Yu Chen.

**Formal analysis:** Yu Chen.

**Funding acquisition:** Yu Chen, Xinqi Gong.

**Investigation:** Yu Chen.

**Methodology:** Yu Chen, Jinping Feng.

**Project administration:** Yu Chen.

**Resources:** Yu Chen.

**Software:** Yu Chen.

**Supervision:** Wei Wang, Xinqi Gong.

**Validation:** Yu Chen, Ying Lu.

**Visualization:** Yu Chen.

**Writing – original draft:** Yu Chen.

**Writing – review & editing:** Yu Chen, Xinqi Gong.

## References

1. Pastor-Satorras R, Vespignani A. Epidemic spreading in scale-free networks. *Physical review letters*. 2001; 86(14):3200. <https://doi.org/10.1103/PhysRevLett.86.3200> PMID: 11290142
2. Wang P, González MC, Hidalgo CA, Barabási AL. Understanding the spreading patterns of mobile phone viruses. *Science*. 2009; 324(5930):1071–1076. <https://doi.org/10.1126/science.1167053> PMID: 19342553
3. Velásquez-Rojas F, Vazquez F. Interacting opinion and disease dynamics in multiplex networks: discontinuous phase transition and nonmonotonic consensus times. *Physical Review E*. 2017; 95(5):052315. <https://doi.org/10.1103/PhysRevE.95.052315> PMID: 28618582
4. Moreno Y, Nekovee M, Pacheco AF. Dynamics of rumor spreading in complex networks. *Physical Review E*. 2004; 69(6):066130. <https://doi.org/10.1103/PhysRevE.69.066130>
5. Domingos P, Richardson M. Mining the network value of customers. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*; 2001. p. 57–66.
6. Kempe D, Kleinberg J, Tardos E. 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. p. 137–146.
7. Carnes T, Nagarajan C, Wild SM, Van Zuylen A. Maximizing influence in a competitive social network: a follower's perspective. In: *Proceedings of the ninth international conference on Electronic commerce*; 2007. p. 351–360.
8. Budak C, Agrawal D, El Abbadi A. Limiting the spread of misinformation in social networks. In: *Proceedings of the 20th international conference on World wide web*; 2011. p. 665–674.
9. Chen W, Collins A, Cummings R, Ke T, Liu Z, Rincon D, et al. Influence maximization in social networks when negative opinions may emerge and propagate. In: *Proceedings of the 2011 siam international conference on data mining*. SIAM; 2011. p. 379–390.
10. Brummitt CD, Lee KM, Goh KI. Multiplexity-facilitated cascades in networks. *Physical Review E*. 2012; 85(4):045102. <https://doi.org/10.1103/PhysRevE.85.045102>
11. He X, Song G, Chen W, Jiang Q. 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. p. 463–474.
12. Li S, Zhu Y, Li D, Kim D, Huang H. Rumor restriction in online social networks. In: *2013 IEEE 32nd international performance computing and communications conference (IPCCC)*. IEEE; 2013. p. 1–10.
13. Shi T, Cheng S, Cai Z, Li Y, Li J. Retrieving the maximal time-bounded positive influence set from social networks. *Personal and Ubiquitous Computing*. 2016; 20(5):717–730. <https://doi.org/10.1007/s00779-016-0943-7>
14. Leskovec J, Huttenlocher D, Kleinberg J. Predicting positive and negative links in online social networks. In: *Proceedings of the 19th international conference on World wide web*; 2010. p. 641–650.
15. Zhu Y, Li D, Yan R, Wu W, Bi Y. Maximizing the influence and profit in social networks. *IEEE Transactions on Computational Social Systems*. 2017; 4(3):54–64. <https://doi.org/10.1109/TCSS.2017.2715183>
16. Tang Y, Shi Y, Xiao X. Influence maximization in near-linear time: A martingale approach. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*; 2015. p. 1539–1554.
17. Granell C, Gómez S, Arenas A. Competing spreading processes on multiplex networks: awareness and epidemics. *Physical review E*. 2014; 90(1):012808. <https://doi.org/10.1103/PhysRevE.90.012808>

18. Jovanovski P, Tomovski I, Kocarev L. Modeling the Spread of Multiple Contagions on Multilayer Networks. arXiv preprint arXiv:170302906. 2017;.
19. Funk S, Jansen VA. Interacting epidemics on overlay networks. *Physical Review E*. 2010; 81(3):036118. <https://doi.org/10.1103/PhysRevE.81.036118>
20. Marceau V, Noël PA, Hébert-Dufresne L, Allard A, Dubé LJ. Modeling the dynamical interaction between epidemics on overlay networks. *Physical Review E*. 2011; 84(2):026105. <https://doi.org/10.1103/PhysRevE.84.026105>
21. Gaye I, Mendy G, Ouya S, Diop I, Seck D. Multi-diffusion degree centrality measure to maximize the influence spread in the multilayer social networks. In: International Conference on e-Infrastructure and e-Services for Developing Countries. Springer; 2016. p. 53-65.
22. Li GL, Chu YP, Feng JH, XU YQ. Influence maximization on multiple social networks [J]. *Chinese Journal of Computers*. 2016; 39:643–656.
23. Bródka P, Musial K, Jankowski J. Interacting spreading processes in multilayer networks. arXiv preprint arXiv:190305932. 2019;.
24. Liu T, Li P, Chen Y, Zhang J. Community size effects on epidemic spreading in multiplex social networks. *PloS one*. 2016; 11(3). <https://doi.org/10.1371/journal.pone.0152021>
25. Eames KT. Networks of influence and infection: parental choices and childhood disease. *Journal of the Royal Society Interface*. 2009; 6(38):811–814. <https://doi.org/10.1098/rsif.2009.0085>
26. Nicosia V, Skardal PS, Arenas A, Latora V. Collective phenomena emerging from the interactions between dynamical processes in multiplex networks. *Physical review letters*. 2017; 118(13):138302. <https://doi.org/10.1103/PhysRevLett.118.138302> PMID: 28409987
27. Guo Q, Jiang X, Lei Y, Li M, Ma Y, Zheng Z. Two-stage effects of awareness cascade on epidemic spreading in multiplex networks. *Physical Review E*. 2015; 91(1):012822. <https://doi.org/10.1103/PhysRevE.91.012822>
28. Zang H. The effects of global awareness on the spreading of epidemics in multiplex networks. *Physica A: Statistical Mechanics and its Applications*. 2018; 492:1495–1506. <https://doi.org/10.1016/j.physa.2017.11.076>
29. Lu W, Bonchi F, Goyal A, Lakshmanan LV. The bang for the buck: fair competitive viral marketing from the host perspective. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining; 2013. p. 928-936.
30. Yu Y, Jia J, Li D, Zhu Y. Fair Multi-influence Maximization in Competitive Social Networks. In: International Conference on Wireless Algorithms, Systems, and Applications. Springer; 2017. p. 253–265.
31. Leskovec J, Sosič R. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*. 2016; 8(1):1–20. <https://doi.org/10.1145/2898361>