Research on Influence Maximization Method for Complex Network

Qipeng Lu[1, *], Pengfei Ding[2]

[1] Nanjing University of Finance and Economics, Nanjing 210000, Jiangsu,China
[2] Nanjin Institute of Technology , Nanjing 211100, Jiangsu, China
[*]Corresponding author e-mail: qipenglu@qq.com

ABSTRACT. *The influence maximization (IM) is a key algorithm problem in information dissemination research. it aims to select a set of K users (also called seed sets) from a network and maximize the number of affected users (influence spread) through a specific information dissemination model. However, despite its huge application potential, with the advent of the era of big data, all kinds of networks tend to be complicated, and there is relatively little research on influence maximization of multilayer networks in complex networks, because in these networks, nodes are different types. On the other hand, most of the existing research on influence maximization relies on greedy algorithms and can only obtain a single solution. With that in mind, we focus on the influence maximization problem of multilayer networks in complex networks. specifically, we first define some novel concepts about the process of information dissemination in multilayer networks; then, we construct the influence maximization problem in multilayer networks into a multi-objective optimization problem. Finally, we do a lot of experiments on the real datasets, and the results show that the algorithm in this paper has a large competitive advantage in the influence spread and running time compared with the existing influence maximization algorithm.*

KEYWORDS: *Influence Maximization, Multilayer Networks, Multi-objective Optimization*

1. Introduction

The rapid development of online social networks makes it easy for people to share information and communicate with each other. Under this background, many large online social network platforms (OSNPs) with billions of users have appeared, such as: Twitter, Facebook and Pinterest. Generally, these OSNPs can be used not only as a communication tool for users, but also as a potential marketing medium for companies and advertisers. A recent study has shown that the word-of-mouth effect stemming from those closely connected social circle of friends can make information disseminates faster and has a wider range of influence. In addition, compared with traditional marketing tools, marketing based on OSNPs has the characteristics of high profit and low investment. Therefore, many companies use OSNPs as a potential marketing medium to promote their new products, services, or innovations. However, because each company’s advertising budget is limited, they must use smaller promotion costs to generate a larger advertising response. A feasible solution is to select some online users who can have a huge impact on the OSNPs as their advertising agents. Formally, we denote this as the influence maximization problem (IMP) in the field of complex network analysis, that is, under a certain influence propagation model, a certain number of seed nodes are mined to maximize the influence propagation range.

Existing information dissemination research has paid considerable attention to IMP. However, with the development of online social networks, social participants are no longer limited to one OSNP, but participate in different OSNPs. In reality, one such a scenario is that a considerable number of people maintain multiple social accounts at the same time, which allows them to spread information among different OSNPs .

We can illustrate this by using social accounts on three different platforms and then spreading the information on the three OSNPs. Once a message is learned by a friend, then the message will be further spread on the three OSNPs. If we focusing only on a single OSNP, the dissemination of information will be inaccurate. Therefore, only considering the influence maximization on a single OSNP will not be able to identify the most influential users, which prompts us in a more complex network system to study the IMP. Because, in a multi-layer network, users' influence is evaluated based on all the OSNPs they participate in. From another perspective, when we try to select several users as seed nodes and spread the influence through them, many solutions may appear that can obtain the greatest influence spread. Interestingly, most of the existing IMP research relies on greedy algorithm strategies, they can only get a single solution, which prompts us to study how to obtain multiple solutions to provide decision-makers with a wide range of options.

In this paper, we focus on the IM problem in multilayer networks in more complex systems. First, we build a multilayer network based on the inter-layer relationships of different networks. Then, we define some novel concepts, such as the reciprocal of the length between the pairs and the influence between the node pairs, these concepts can intuitively reveal the information propagation process in the multilayer network. Finally, we describe the IMP in the multilayer network as a multi-objective optimization problem. Specifically, we construct a multi-objective optimization model that fully considers the centrality and information dissemination ability of the candidate seed nodes. In order to solve the target problem, genetic algorithms are used to explore the wide search space of all possible seed node sets. The goal of the genetic algorithm is to find a set of optimal solutions in one run and provide a series of choices for decision makers. Therefore, we propose an IM algorithm based on Nondominated Sorting Genetic Algorithm)(NSGA-II). In order to maintain the diversity of the population and accelerate the convergence of the algorithm, we will combine an effective crossover operation and a gain-based mutation operator.

The rest of this paper is organized as follows: Section 2 briefly describes related work on IMP. Section 3 defines and considers the IMP in multilayer networks. Section 4 details our approach. In Section 5, we thoroughly evaluate our approach and its performance using many experiments. Finally, we conclude in Section 6.

2. Related Work

With the development of the Internet and smart devices, IM research in social networks has gradually become a hot research topic [1]. Researchers are trying to figure out how to maximize the spread of influence in complex networks by mining a certain number of seed nodes. In the existing literature, many concerns have focused on this issue, which follows different strategies, as elaborated below:

IMP was first proposed and studied by Domingos and Richardson et al. [2,3], and then Kempe et al. transformed IMP into a discrete optimization problem, and proved that IM is a NP-hard problem [4]. To effectively solve this problem, Kempe et al. proposed three widely used information dissemination models, they are independent cascade model (IC), linear threshold model (LT) and weight cascade model (WC), and then proposed an approximate ratio of (1-1 /e) greedy algorithm; because this greedy algorithm requires enough Monte Carlo simulations by the information dissemination model to obtain accurate estimates of the average impact propagation, therefore, as the network continues to expand, this application of greedy algorithms is limited and it is difficult to meet large-scale social networks. So, how to reduce the calculation time of the algorithm and improve the calculation performance has become a key challenge.

In order to improve the efficiency of the greedy algorithm, some new greedy algorithms have been proposed. Leskovec et al. [5] proposed a lazy greedy algorithm Cost-Effective Lazy Forward (CELF) by mining the submodeling of the influence function, which greatly reduced the number of simulations to evaluate the seed influence range. The experiment shows that the CELF algorithm is 700 times faster than the greedy algorithm. Although the computing performance of the CELF algorithm has been greatly improved, it still takes several hours to find top-50 seed nodes on a network with thousands of nodes. In addition, Goyal et al. [6] proposed CELF++ to further optimize CELF; experimental results show that CELF++ improves performance by nearly 35~55 percent compared to CELF. Although these improved greedy algorithms improve the runtime, they have poor scalability for large-scale networks.

To address the scalability issue, Chen et al. proposed several heuristic methods, including Degree Discount [7] and PMIA [8], to approximate the influence propagation using each node's local arborescence structures. Jung et al. [9] proposed the IRIE algorithm, which integrates the advantages of influence ranking (IR) and influence estimation (IE) methods for IM. Although these heuristic algorithms are quite efficient, their accuracy can be much lower than a greedy algorithm. As we mentioned previously, because the IM is NP-hard, these methods based on greedy and heuristic algorithms cannot efficiently find promising solutions; instead, they find the optimal set only under certain conditions and some level of approximation.

In recent years, some approaches attempted to tackle the IM problem by means of computational intelligence, exploiting methods such as Simulated Annealing [10] and Evolutionary Algorithms [11, 12]. Evolutionary algorithms were found capable of effectively exploring the vast search space of all possible subsets of nodes. In particular, the genetic algorithms do not require any assumptions about the graph underlying the network, and more feasible solutions are available in these algorithms than current heuristics. These approaches show some promising results on relatively large datasets obtained from real-world networks.

3. Notation definition and problem formulation

The multilayer network in this paper is represented as a two-tuple, i.e., **M=<V,W>**, where $\mathbf{V} = \{i: i \in \{1, \cdots, n\}\}$ is the set of n entities. $\mathbf{W} = \{W^{\alpha}: \alpha \in \{1, \cdots, L\}\}$ is the family of L weighted adjacency matrices, where each element, i.e., $W^{\alpha} = (w_{i,j}^{\alpha}) \in \mathbb{R}^{n \times n}$, represents the directed and weighted network without self-loops on the αth layer of **M.** In this paper, $\forall \alpha \in \{1, \cdots, L\}$ and $\forall i, j \in \mathbf{V}$, we assume that $w_{i,j}^{\alpha} \in [0,1]$, representing the probability that entity *i* transmits the information upon contacting with entity *j* on the α th layer. In contrast with the classic mathematical framework defined by Domenico et al., the inter-layer spreading processes in **M** are created only by the same entity, who has participated in two different layers. In other words, a particular piece of information can spread from one layer to another, through the same entity. To emphasize entity $i's$ position on the α layer, we sometimes represent it as i^{α}. Moreover, $\forall \alpha, \beta \in \{1, \cdots L\}$, $\alpha \neq \beta$ and $\forall i \in \mathbf{V}$, the inter-layer spreading probability from i^{α} to i^{β} is fixed to be 1. We also comprehend that each entity in **V** has the ability of context-awareness on multiple layers of **M**. In a realistic setting, a particular entity *i* can be viewed as a real person; note that *i* can have multiple social network accounts, such as Facebook and Twitter. If *i* receives a piece of information from his Facebook friend *j*, *i* can send the same information to his other friend *k* on Twitter.

Understanding how information spreads in multilayer networks [13] is an important problem, having implications for both predicting the size of epidemics, as well as for planning effective interventions. One of the important ideas with regard to the spreading processes in multilayer networks is that information can also spread from one layer to another. The set of spreading paths from entities *i* to *j* over the multilayer network **M** (denoted by $SetPath_{i \to j}$) is defined as

$$SetPath_{i \to j} = \{\left(v_0^{\beta_0}, v_1^{\alpha_1, \beta_1}, \cdots, v_{\chi}^{\alpha_{\chi}}\right)|_{i=v_0 \wedge j=v_{\chi}}:$$

$$\forall \tau \in \{0, \cdots, \chi\}, \forall_{\tau} \in \mathbf{V},$$

$$\forall \tau, \tau' \in \{0, \cdots, \chi\}, \ v_{\tau} \neq v_{\tau'},$$

$$\forall \tau \in \{1, \cdots, \chi\}, \ \beta_{\tau-1} = \alpha_{\tau} \wedge w_{v_{\tau-1}, \ v_{\tau}}^{\alpha_{\tau}} > 0\} \qquad (1)$$

Where $w_{v_{\tau-1},\ v_\tau}^{\alpha_\tau}$ indicates the probability that entity $v_{\tau-1}$ transmits the information upon contacting his/her out-neighbor entity v_τ on the α_τth layer. Here, the spreading path (if any) from entities *i* to *j* must be acyclic, and each entity $v_\tau(\tau \in \{1,\cdots,\chi\})$ on such a path can be chosen as a random out-neighbor of entity $v_{\tau-1}$ on any arbitrary layer. Thus, this path's probability is given by

$$pro_{i\to j}\left(v_0^{\beta_0}, v_1^{\alpha_1,\beta_1}, \cdots, v_\chi^{\alpha_\chi}\right) = \prod_{\tau=1}^{\chi} w_{v_{\tau-1},\ v_\tau}^{\alpha_\tau} \tag{2}$$

where χ denotes the path's length.

Next, we will introduce some novel measures to evaluate the information spreading ability from entities *i* to *j* over the multilayer network **M**:

Definition 1. Pairwise Reciprocal Length: The pairwise reciprocal length from *i* to *j* is defined as the reciprocal of the minimal length of all spreading paths from *i* to *j*:

$$rel_{i\to j} = \max_{\left(v_0^{\beta_0}, v_1^{\alpha_1,\beta_1}, \cdots, v_\chi^{\alpha_\chi}\right) \in SetPath_{i\to j}} \frac{1}{\chi} \tag{3}$$

Definition 2. Pairwise Influence: The pairwise influence of *i* on *j* is defined as the maximal probability of all spreading paths from *i* to *j*:

$$inf_{i\to j} = \max_{\left(v_0^{\beta_0}, v_1^{\alpha_1,\beta_1}, \cdots, v_\chi^{\alpha_\chi}\right) \in SetPath_{i\to j}} pro_{i\to j}(v_0^{\beta_0}, v_1^{\alpha_1,\beta_1}, \cdots, v_\chi^{\alpha_\chi}) \tag{4}$$

Broadly speaking, the higher the values of $rel_{i\to j}$ and $inf_{i\to j}$, the shorter the distance and the greater the influence, from entities *i* to *j* over the multilayer network **M**, respectively. Note that if there is no spreading path from *i* to *j* (i.e., $SetPath_{i\to j} = \emptyset$), we have $rel_{i\to j} = inf_{i\to j} = 0$.

Given a multilayer network **M=<V,W>**, let S⊂**V** denote a seed set of K entities(e.g., |S|=K), some notions of the information spreading ability associated with S are defined as follows:

Definition 3. Harmonic Centrality: The Harmonic centrality of S is the sum of the maximal pairwise reciprocal length from the entities in S to each other entity in **V**/S:

$$\mathrm{H(S)} = \sum_{j\in V/S} \max_{i\in S} rel_{i\to j} \tag{5}$$

Definition 4. Accessibility: The accessibility of S is the sum of the maximal pairwise influence from the entities in S to each other in **V**/S:

$$\mathrm{A(S)} = \sum_{j\in V/S} \max_{i\in S} inf_{i\to j} \tag{6}$$

In contrast with the conventional influence maximization problem, which aims to select K entities so that the expected number of entities influenced by these K entities will be maximized. In this paper, the problem of influence maximization is defined as the following multi-objective optimization problem:

$$\max_{S\subset V} Q(S) = \left(Q^1(S), Q^2(S)\right)^T,$$

$$Q^1(S) = \mathrm{H}(S) = \sum_{j\in V/S} \max_{i\in S} rel_{i\to j}$$

$$Q^2(S) = \mathrm{A}(S) = \sum_{j\in V/S} \max_{i\in s} inf_{i\to j}$$

$$\mathrm{s.t.}, |S| = \mathrm{K}. \tag{7}$$

In general, as the two objective in Eq.(7) are often conflicting with each other, it is very hard or impossible to find a single solution S^* that optimizes all objectives simultaneously. An alternative feasible solution is to find a good balance among the multiple objectives so that each one has a relatively satisfied value. Such solutions are also called the Pareto optimal solutions in the filed of multi-objective optimization. In this paper, we will introduce a novel evolutionary algorithm(IMA-MOEA) based on the classic Non-dominated Sorting Genetic Algorithm framework to find the set of Pareto optimal solutions. The influence maximization problem as defined in Eq.(7).

4. Methodology

4.1 The multi-side projection network

For a given multilayer network **M=<V,W>,** the multi-side projection network of **M** is defined as proj(**M**)=$< W^H, W^A >$, where $W^H = (w^H_{i,j}) \in \mathbb{R}^{n\times n}$ and $W^A = (w^A_{i,j}) \in \mathbb{R}^{n\times n}$ are the two n × n weighted adjacency matrices, such that $\forall i, j \in$ **V**:

$$w^H_{i,j} = \begin{cases} 1 \; \exists \alpha \in \{1,2,\cdots,L\} \Rightarrow w^\alpha_{i,j} > 0 \\ 0 \; otherwise \end{cases} \tag{8}$$

$$w^A_{i,j} = \max_{\alpha\in\{1,2,\cdots,L\}} w^\alpha_{i,j} \tag{9}$$

It is important to remark that the two weighted adjacency matrices, i.e., W^H and W^A, associated with the projection network proj(**M**) describe the topological characteristics of **M** from the two different perspectives. Along the line, the sets of out-neighbors and in-neighbors associated with entity *i* on the projection network proj(**M**) can be defined as $\widehat{N}_i^{proj} = \{j: w^H_{i,j} > 0\}$ and $\widetilde{N}_i^{proj} = \{j: w^H_{j,i} > 0\}$,respectively. Furthermore, the set of spreading paths from entities *i* to *j* over the projection network (denoted by $SetPath^{proj}_{i\to j}$) can be defined as

$$SetPath_{i\to j}^{proj} = \{(v_0, v_1, \cdots, v_\chi)|_{i=v_0 \wedge j=v_\chi})$$
$$\forall\tau \in \{0, \cdots, \chi\}, \forall_\tau \in \mathbf{V},$$
$$\forall\tau, \tau' \in \{0, \cdots, \chi\}, \ v_\tau \neq v_{\tau'},$$
$$\forall\tau \in \{1, \cdots, \chi\}, v_\tau \in \widehat{N}_{v_{\tau-1}}^{proj}\}, \tag{10}$$

Where each path should be also simple and acyclic. Based on the definition of $SetPath_{i\to j}^{proj}$, one can easily deduce the following two properties:

Property 1. The pairwise reciprocal length from *i* to *j* over the multilayer network (see **Definition 1**) can be seen as the reciprocal of the shortest path length from *i* to *j* over the corresponding projection network, such that

$$rel_{i\to j} = \max_{(v_0,v_1,\cdots,v_\chi)\in SetPath_{i\to j}^{proj}} \frac{1}{\chi} \tag{11}$$

Property 2. The pairwise influence of *i* on *j* over the multilayer network (see **Definition 2**) is actually the maximal probability of all spreading paths from *i* to *j* over the corresponding projection network, such that

$$inf_{i\to j} = \max_{(v_0,v_1,\cdots,v_\chi)\in SetPath_{i\to j}^{proj}} \prod_{\tau=1}^{\chi} w_{v_{\tau-1},\ v_\tau}^{A} \tag{12}$$

4.2 Chromosome representation

In the multi-objective evolutionary algorithms, the solution is encoded as a chromosome firstly, and how to represent the chromosome is very important in the evolutionary algorithms. It not only has an effect on the selection of population diversity, but also on the efficiency of evolutionary algorithms. A chromosome with n (number of entities in a multilayer network) genes is represented as $\mathrm{x} = (x_1, x_2, \cdots, x_n)$,where every $\mathrm{x} \in \Omega$ is a *n*-dimensional vector of decision variables, $\Omega = \{0,1\}^n$ is the solution space, and each variable x_i is the value of a gene. In this paper, we take binary representation for the value of each gene. If the value of gene $x_i = 1$, it means that entity *i* is existed in the set of seeds S, on the contrary, if the value of gene $x_i = 0$, it means that entity *i* is not existed in the set of seeds S. Obviously, the number of 1 in a chromosome is equal to the size of K. Based on the definition of chromosome, the next two properties can be deduced easily.

Property 3. The Harmonic centrality of x is the sum of the maximal pairwise reciprocal length from the genes in $\mathrm{C}_1(\mathrm{x})$ to each other gene in $\mathcal{C}_0(x)$:

$$\mathrm{H(x)} = \sum_{j\in\mathcal{C}_0(x)} \max_{i\in\mathcal{C}_1(x)} rel_{i\to j} \tag{13}$$

Property 4. The accessibility of x is the sum of the maximal pairwise influence from the genes in $\mathrm{C}_1(\mathrm{x})$ to each other gene in $C_0(x)$:

$$\mathrm{A(x)} = \sum_{j\in C_0 x} \max_{i\in C_1(x)} inf_{i\to j} \tag{14}$$

By its very nature, the problem of multi-objective optimization problem(MOP) can be redefined as:

$$\max_{\mathrm{x}\in\Omega} Q(x) = (Q^1(x), Q^2(x))^T,$$

$$Q^1(x) = \mathrm{H(x)} = \sum_{j\in C_0(x)} \max_{i\in C_1(x)} rel_{i\to j}$$

$$Q^2(x) = \mathrm{A(x)} = \sum_{j\in C_0(x)} \max_{i\in C_1(x)} inf_{i\to j}$$

$$\mathrm{s.t.}, \forall i,j \in \{1,\cdots,n\}, \mathrm{x}\in\Omega,\quad \forall k \in \{0,1\}, C_k(x) = \{i | x_i = k\} \tag{15}$$

The $x_i = \{0,1\}$ is the strategy space of each gene i, and the $C_k(x)$ represent a set of genes when $x_i = k$. In this context, we are aimed to find the set of Pareto optimal solutions, which should keep a diversity of solutions.

4.3 Genetic operators

4.3.1 Crossover operator

The purpose of crossover and mutation in evolutionary algorithms is to speed up the evolution of population in order to generate new chromosomes. Crossover is the process of exchanging parts of genes from two parent chromosomes to produce new chromosomes, and it is one of the key factors in natural biological evolution. Traditional crossover operators, such as one-point crossover, partial-mapped crossover, cycle crossover and uniform crossover, are not suitable for our algorithm, because in our approach, each chromosome is represented by binary, and the label of each gene locus can be well translated into an entity's label. Therefore we must ensure that the number of genes with the same value in each chromosome is the same. Next, we will introduce a novel crossover operation, which can satisfy the constraint that the number of gene values equal to 1 is K. First, select two chromosomes x_1 and x_2 as paternal chromosomes according to binary tournament selection. Then randomly pick two genes *i* and *j* in chromosome x_1 whose values are one and zero, respectively. Ensure that the allele values in chromosome x_2 are zero and one. Simultaneously, exchange of alleles on two chromosomes $x_1$and x_2. Thus, two new chromosomes x_3 and x_4 have been generated. Next, we calibrate each gene of the newly generated chromosomes to ensure that each newly generated chromosome conforms to the representation of chromosome.

4.3.2 Gain-Based mutation operator

Crossover can transfer good genes from parent generation to the next generation and make offspring superior to parent generation. However, premature convergence occurs when the offspring of the cross generation are not as good as the parent generation. The reason is the occurrence of an effective gene deletion. To overcome this situation, a mutation operation is used. The traditional mutation operation is to randomly change one or more gene sites in a chromosome. The probability of change is called mutation probability p_m. In simple genetic algorithm, mutation is a random change of the value of a gene in a chromosome with probabilistic p_m, that is, a simple transformation of one to zero or vice versa at a particular position. This traditional mutation operator is always random and we cannot guarantee to generate better solutions. Thus, it is necessary to apply a heuristic algorithm to mutation operation. Focus on the issues mentioned above, a gain-based mutation is designed in our algorithm. The essence of this mutation operation is to remove one of the worst nodes based on one objective and then update the chromosomes based on the maximum gain of two objectives, respectively. First, select a chromosome $\mathrm{x} = [x_1, x_2, \cdots, x_n]$ by the binary tournament selection and remove one gene/entity with the minimum gain based on each objective. The gain is defined as the increased objective value after gene/entity i joins the set S. The specific expression is as follows:

$$\mathrm{gain}(i|\mathrm{S}, k) = Q^k(\mathrm{S} \cup i) - Q^k(\mathrm{S}) \tag{16}$$

$k \in \{1,2\}$, which represents the number of the objective. Second, adding a gene/entity maximizes the gain based on each objective so that we can get two new chromosomes x_1, x_2. Then, generating a vector $\mathrm{x}^{'}$ by summating the alleles of the two newly generated chromosomes. If the values on the corresponding position of vector x' is equal or over 1, it means that the genes in the corresponding position are not the worst of at least one objectives, these genes will eventually be retained. In order to maintain the size of $x_i = 1$ in the mutation chromosome is equal to |S|, we select the top K as the reserved genes based on the value at the corresponding position in vector x'. So, a new mutation chromosome $x''$is created. A simple example for how to generate a new mutation chromosome based on the gain is shown following. For example, we select a chromosome $\mathrm{x} = [0,1,0,0,1,0,1,1,0,0]$, removing a gene 7 with a minimum gain based on objective one and adding a gene 3 with a maximum gain based on objective one, in this way, a new chromosome $\mathrm{x}_1 = [0,1,1,0,1,0,0,1,0,0]$is generated. In the same way, removing a gene 7 with a minimum gain based on objective two and adding a gene 4 with a maximum gain based on objective two, chromosome $\mathrm{x}_2 = [0,1,0,1,1,0,0,1,0,0]$ can be obtained. A vector $\mathrm{x}^{'} = [0,2,1,1,2,0,0,2,0,0]$ by summating the alleles of chromosome x_1 and x_2 can be got. Then, according to the value at each position of the vector $\mathrm{x}^{'}$, the top K(assume K=4) are selected as the reserved genes and assign value 1 to the corresponding positions, other genes are assigned to 0. Finally, a new mutation chromosome $\mathrm{x}^{''} = [0,1,1,0,1,0,0,1,0,0]$ is created.

As mentioned above, we use gain-based operator to update a chromosome. It is proved that this method can accelerate the selection of optimal solution.

4.4 The idea of IMA-MOEA

In the past few years, researchers have been working on solving MOPs with EAs. NSGA-II[14] is one of the best genetic algorithms for solving MOPs. To uncover how information is spread across multilayer networks, as well as to reconcile the operational efficiency and the computational complexity, in this work, a new evolutionary algorithm IMA-MOEA is introduced into the classical NSGA-II framework to solve the influence maximization in Eq.(15). In IMA-MOEA, each chromosome stands for a seed set and each population contains N chromosomes. IMA-MOEA starts from with a set of chromosomes selected by the binary tournament, and then ,uses the above two objectives $\mathrm{Q}^1(\mathrm{x})$ and $\mathrm{Q}^2(\mathrm{x})$ to evaluate the quality of each chromosome, next, performs a series of evolutionary operators such as crossover and gain-based mutation on the chromosome to generate a new offspring population which is used in the next population evolution. Repeat the above operations until the termination condition is met.

The specific details of MOEA will be shown in algorithm 1.

Algorithm 1: The general framework of IMA-MOEA

Input: A given multilayer network $\mathbf{M} =< \mathbf{V}, \mathbf{W} >$ and the maximum number of iterations T_s;

Output: The Pareto optimal set $\boldsymbol{PS}$;

```
$t \leftarrow 0$;
$P_t \leftarrow$ population initialization($\mathbf{M}$)
$\boldsymbol{F} \leftarrow$ fast non-dominated sorting ($\boldsymbol{P}_t$)
$\forall \boldsymbol{F_i} \in \boldsymbol{F}$, crowing distance assignment ($\boldsymbol{F_i}$);
repeat
    $\boldsymbol{Q}_t \leftarrow \emptyset$;
    repeat
        $[\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3] \leftarrow$ binary tournament selection ($\boldsymbol{F}$);
        $\mathrm{x}_1^{''} \leftarrow$ gain-based mutation x;
        $[\mathrm{x}_2^{'}, \mathrm{x}_3^{'}] \leftarrow$ crossover operator ($\mathrm{x}_2, \mathrm{x}_3$);
        $\mathrm{Q_t} \leftarrow \mathrm{Q_t} \cup \{\mathrm{x}_1^{''}, \mathrm{x}_2^{'}, \mathrm{x}_3^{'}\}$;
    until $|\boldsymbol{Q}_t| < |\boldsymbol{P}_t|$
    $\boldsymbol{R_t} \leftarrow \boldsymbol{P_t} \cup \boldsymbol{Q_t}$ ;
    $\boldsymbol{F} \leftarrow$ fast non-dominated sort ($\boldsymbol{R}_t$);
    $\boldsymbol{P}_{t+1} \leftarrow \emptyset$ and $i \leftarrow 1$;
    repeat
        crowding distance assignment($\boldsymbol{F_i}$);
        $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup \boldsymbol{F}_i$;
        $i \leftarrow i + 1$;
    until $|\boldsymbol{P}_{t+1}| + |\boldsymbol{F}_i| \leq |\boldsymbol{P}_t|$
    crowding distance assignment($\boldsymbol{F}_i$);
    sort $\boldsymbol{F}_i$ in descending order of the crowing distance;
    $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup \boldsymbol{F}_i[1:(|\boldsymbol{P}_t| - |\boldsymbol{P}_{t+1}|)]$;
    $t \leftarrow t + 1$;
until $t < T_s$
return $\boldsymbol{P}_t$;
```

5. Experiments

In order to evaluate the IMA-MOEA algorithm, this section conducts experiments on datasets such as Network Science (Nets), high-Energy Theory(Het) and Astrophysics Physics Collaborations (Ac), and compares the algorithm proposed in this paper with several current advanced algorithms. We evaluated the algorithm performance difference in terms of influence spread and running time. The test environment of this experiment was Inter Core i7 CPU@3.6GHz 16GB and all datasets came from professor Newman's personal data website (http://www-personal.umich.edu).

5.1 Experimental setup

The analysis of influence maximization in a multilayer network needs to consider the connection relationship and entity recognition problems between entities on different networks. Although there have been some research results to solve the entity connection in the multilayer networks, considering the accuracy problem, we will use entities to correspond to one-to-one datasets Network Science (Nets), High-Energy Theory (Het) and Astrophysics collaborations (Ac). These datasets are weighted and directed scientific research cooperation networks, each node in the network has a real name. In this paper, nodes with the same name refer to the same entity. Node information on different networks is identified by the node name and network ID. Therefore, for the construction of a multilayer network, it can be done through multiple cooperative networks. Specific in particular, the inter-layer connection of a multilayer network dataset Nets-Het can only be established through the same entity in both Nets and Het networks. Using the same method, two other multilayer network dataset Nets- Ac and Het- Ac are created, of which 89 co-authors in the Nets and Het networks, 90 co-authors in the Nets and Ac networks, and 1,290 co-authors in the Het and Ac networks. In order to facilitate the calculation, we map the multilayer network into a projection network and maximize the weight of the edges in the multilayer network. In addition, in order to more accurately reveal the process of information dissemination, we extract maximum connected subgraphs from each dataset to study the issue of maximizing influence. Some basic statistics about the datasets are shown in Table 1.

Table 1 Datasets' basic statistics

Datasets	Node	Edge	Average degree	Average clustering coefficient
Nets	1589	2742	3.451	0.319
Het	8361	15751	3.768	0.221
Ac	16706	121251	14.516	0.726

In this paper, we compare the IMA-MOEA with several classic algorithms, including the Greedy, High Degree [15], Degree Discount [7], PageRank [16] and LDAG [17]. The following is a list of algorithms we evaluate in our experiments.

- **LDAG**: This algorithm is designed for the LT model. We use the influence parameter $\theta = \frac{1}{320}$ to control the local DAG's size constructed for each node.

- **Degree Discount**: This is a basic degree discount heuristic algorithm applicable to all cascade models, with a propagation probability of *p*=0.1. Although this heuristic is designed specifically for the independent cascade model, we use it as a general heuristic in the class of degree heuristics. It performs much better than the pure-degree heuristics.

- **High Degree**: As a kind of comparison algorithm, High Degree is a simple heuristic algorithm for selecting seed sets. The main idea is that the higher the degree, the more influential the node. It selects top-*k* nodes with the highest out-degree as a seed set.

- **Greedy**: This is a greedy algorithm that uses lazy-forward optimization [15]. To obtain an accurate estimate of influence spread, we run 10,000 simulations.

- **PageRank**: This popular algorithm rank webpages. The weight $w_{j,i}$ on the edge from entity *i* to entity *j* indicates the transition probability. Intuitively, $w_{j,i}$ indicates the influence strength of entity *j* to entity *i*, and we use it in the opposite direction as a "vote" of entity *i* to entity *j*, as explained by the PageRank algorithm. *K* nodes with the highest PageRanks will be selected as seed sets. We set the restart probability for PageRank as 0.15, and the stop criterion as 0.0001 in L_1 norm.

In order to further evaluate the robustness of the proposed algorithm and the accuracy of the comparison method, we will calculate the influence spread of the final solution set of each algorithm on the LT propagation model. Based on the above work, some parameters of the proposed algorithm IMA-MOEA will be set as T_s=500 and K value from 10 to 50.

5.2 Experimental results

5.2.1 Influence spread for real-world datasets

Because the algorithm we propose is an evolutionary algorithm, a set of solutions can be obtained for different sizes of seed sets. In order to facilitate the comparison between different seed solutions, we will choose one who has a great influence on seed solution set, it also embodies the advantages of the algorithm proposed in this paper, that is, it can generate many solutions in a single run, providing a wide range of strategies for decision makers. In this experiment, we set the size of the seed set K to be from 10 to 50 . For ease of reading, the percentage difference of all the influence spread in the following content is taken K=50 as an example.

In order to verify the effectiveness and robustness of the proposed algorithm, we conducted a large number of experiments on the basis of the LT model. In the LT model, the activation threshold of an entity is crucial to influence propagation, so a threshold strategy that will be adopted for the threshold of an entity is that the activation threshold of all entities is a random number between (0, 1). Figures 1, 2 and 3 show the comparison of the influence spread of several methods on three different datasets when the activation threshold of each entity is random. In figure 1, an obvious phenomenon is LDAG algorithm is better than Greedy, Degree Discount, High Degree, and Pagerank algorithms, but it is still worse than the IMA-MOEA algorithm proposed in this paper. This is because the LDAG algorithm is specifically designed for the LT model and has better performance on the LT model. In addition, with the increase in the number of seeds K, the growth rate of the influence spread of the IMA-MOEA algorithm and the LDAG algorithm changes slowly, but it is still better than other comparison algorithms, which provides new ideas for decision makers to select seed, that is, under the LT model, the strategy of seed selection should first consider IMA-MOEA algorithm and LDAG algorithm. In Figure 2, the IMA-MOEA algorithm performs better than all other algorithms. It performs 12.99%, 56.12%, 56.55%, and 33.85% higher than the performance of LDAG, Degree Discount, High Degree, and Greedy algorithms. Degree Discount, High Degree, and Greedy have similar influence spread, and they are similar to the performance of the LDAG algorithm. The Pagerank algorithm has very poor performance on any dataset, and completely loses its competitiveness compared with other algorithms. As can be seen in figure 3, the performance of the Pagerank algorithm on the Het-Ac dataset is greatly improved compared to the other two datasets, but it is still the worst performance. From the above results of the activation thresholds of all entities taking random values between $\theta \in (0,1)$, the experimental results of influence maximization in the multilayer networks show that the IMA-MOEA algorithms has a better performance and the algorithm effect is more prominent. This also indicates that when the entity activation threshold is taken into account in the selection strategy of seed entities, the change of the initial entity activation threshold will not cause the performance degradation of this algorithm.

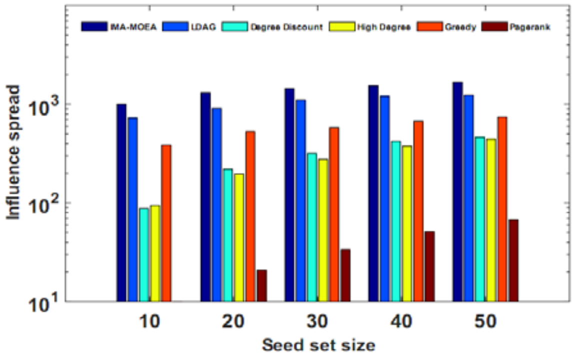

Figure 1. Various algorithms' influence spread under the linear threshold model in the Nets-Het dataset.

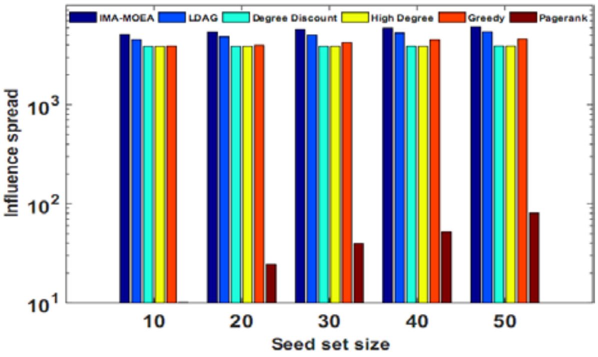

Figure 2. Various algorithms' influence spread under the linear threshold model in the Nets-Ac dataset.

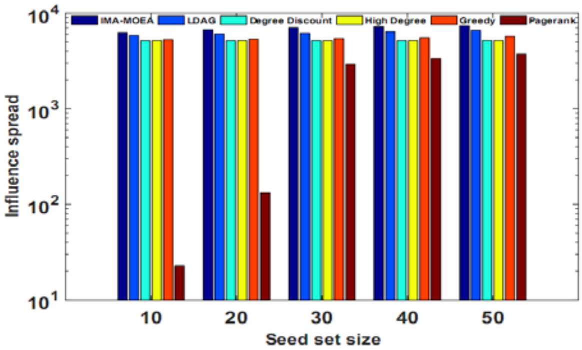

Figure 3. Various algorithms' influence spread under the linear threshold model in the Het-Ac dataset.

5.2.2 Running time for real-world datasets

In order to verify the difference between the IMA-MOEA algorithm and other algorithms in terms of computing efficiency, we also conducted a large number of experiments to compare the running time of the IMA-MOEA algorithm and other algorithms on three real datasets. Figure 4 shows the running time of selecting 50 seed entities by various algorithms on three real datasets. Intuitively, when the size of the dataset increases, the running time of the Greedy algorithm will also increase and it is the most time-consuming algorithm to calculate, whether it is on the Nets-Het, Nets-Ac, or Het-Ac datasets. In addition, the algorithm IMA-MOEA proposed in this paper can select 50 seed entities within a stable calculation time on any dataset. Specifically, IMA-MOEA takes only a few seconds to select 50 seed entities on three datasets, which is three to four orders of magnitude faster than the traditional greedy algorithm. Another phenomenon that can be observed is that in the three datasets, the IMA-MOEA algorithm requires a greater cost than the LDAG algorithm in terms of calculation time. This is because the IMA-MOEA algorithm searches a wide range of all possible subsets of nodes and can find a set of optimal solutions in a single run, which is very time-consuming. In contrast, other algorithms (Degree Discount, High Degree and Pagerank) have shorter running times, and they are all better than IMA-MOEA. In terms of running time, High Degree has the best performance, but it cannot find a high quality seed set. The greedy algorithm can choose a more reliable solution, but with the continuous expansion of the network scale and the complexity of the network relationship, the algorithm shows poor scalability. Therefore, it can be concluded that IMA-MOEA and LDAG algorithms have good efficiency both in the influence spread and in running time, and IMA-MOEA algorithm balances good efficiency and reliable solutions, which is the best choice to seek more feasible solutions in one run.

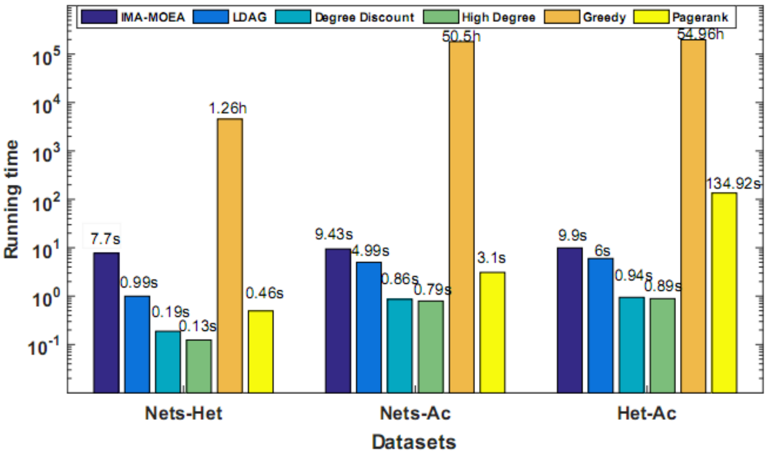

Figure 4. Running time of different algorithms on the three datasets.

6. Conclusion

This paper studies the problem of influence maximization on multilayer networks in complex networks and proposes a multi-objective evolution method IMA-MOEA for influence maximization in multilayer networks. Specifically, compared with a single network, a multilayer network has the characteristics of entity self-propagation and more complicated relationships. Therefore, we first use the propagation characteristics of the entities to connect multiple networks to build a multilayer network; then we propose some novel concepts for information propagation in the multilayer networks and construct the problem of influence maximization into a multi-objective optimization problem ; Then, under the linear threshold influence model, the problem of influence maximization in multilayer networks is solved. Finally, we conducted a lot of experiments on three real datasets. From the experimental data, it can be seen that the IMA-MOEA algorithm performs better than other algorithms or at least as well as the best tested algorithm on each dataset. In general, the experiments in this paper show that evolutionary algorithms are a feasible tool for solving the problem of influence maximization, especially in the case of seeking multiple feasible solutions.

References

[1] Doina Bucur, Giovanni Iacca and Andrea Marcelli, et al (2018). Improving Multi-objective Evolutionary Influence Maximization in Social Networks. In International Conference on the Applications of Evolutionary Computation. p.117–124.

[2] Domingos P, Richardson M (2001). Mining the network value of customers//Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p.57-66.

[3] Richardson M, Domingos P (2002). Mining knowledge-sharing sites for viral marketing//Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p.61-70.

[4] Kempe D, Kleinberg J, Tardos E (2003). Maximizing the spread of influence through a social network//Proceeding of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p.137-146.

[5] Leskovec J, Krause A, Guestrin C, et al (2007). Cost-effective outbreak detection in networks//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p.420-429.

[6] Goyal A, Lu W, Lakshmanan L V (2011). CELF++: Optimizing the greedy algorithm for influence maximization in social networks//Proceedings of the 20th International Conference Companion on Word Wide Web. p.47-48.

[7] Wei Chen, Yajun Wang, and Siyu Yang (2009). Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. p.199–208.

[8] Wei Chen, Chi Wang, and Yajun Wang (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. P.1029–1038.

[9] Jung K, Heo W, Chen W (2012). IRIE: Scalable and robust influence maximization in social networks//Proceedings of the 2012 IEEE 12th Internation Conference on Data Mining. Brussels，Belgium，p.918-923.

[10] Jiang Q, Song G, Gao C, et al (2011). Simulated annealing based influence maximization in social networks//Proceedings of the 25th AAAI Conference on Artificial Intelligence. California, p.127-132.

[11] Bucur D, Iacca G, Marcelli A, et al (2018). Improving multi-objective evolutionary influence maximization in social networks. Applications of Evolutionary Computation, p.117-124.

[12] Weskida M, Michalski R (2016). Evolutionary algorithm for seed selection in social influence process//Peoceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. San Francisco, p.1189-1196.

[13] Salehi M, Sharma R, Marzolla M, et al (2015). Spreading processes in multilayer networks. IEEE Transactions on Network Science and Engineering, p.65-83.

[14] Deb K, Pratap A, Agarwal S，et al (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, p.182-197.

[15] E. Tardos, D. Kempe, and J. Kleinberg (2003). Maximizing the spread of influence in a social network. In ACM SIGKDD international conference on knowledge discovery and data mining. P.137–146.

[16] Brin S, Page L (1998). The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems, p.107-117.

[17] Chen W, Yuan Y, Zhang L (2010). Scalable influence maximization in social networks under the linear threshold model//Proceedings of the 10th IEEE International Conference on Data Mining. p.88-97.