Abstract

Online social networks play an important role in marketing services. Influence maximization is a major challenge given its goal to find the most influential users in a social network. Increasing the number of influenced users at the end of a diffusion process while decreasing the time of diffusion are two main objectives of the influence maximization problem. The goal of this paper is to find multiple sets of influential users such that each of them is the best set to spread influence for a specific time bound. Considering two adverse objectives, increasing influence and decreasing diffusion time, we employ the NSGA-II algorithm which is a powerful algorithm in multi-objective optimization to find different seed sets with high influence at different diffusion times. Since social networks are large, computing influence and diffusion time of all chromosomes in each iteration will be challenging and computationally expensive. Therefore, we propose two methods which can estimate the expected influence and diffusion time of a seed set in an efficient manner. Providing the set of all potentially optimal solutions, help a decision maker evaluate the tradeoffs between the two objectives, i.e., the number of influenced users and diffusion time. In addition, we develop an approach for selecting seed sets, which have optimal influence for specific time bounds, from the resulting Pareto front of the NSGA-II. Finally, we show that applying our algorithm to real social networks outperforms existing algorithms for influence maximization problem. The results show a good compromise between two objectives and the final seed sets result in high influence for different time bounds.

Keywords: Diffusion time; Influence maximization; Multi-objective optimization; Social networks; Trade-off decision making.

1. Introduction

Millions of people around the world subscribe to popular social networks such as Facebook, twitter, google+ and other social media. The high usage of social networks provides a good opportunity for social network analysis. There are many different research directions in social network analysis, including information diffusion [1–3], community detection [4–7], link prediction [8–10] and behaviour analysis [11,12].

As mentioned before, one of the main research interests in social network is analysing information diffusion. Information diffusion is an important mechanism for viral marketing. Viral marketing is a marketing strategy that is based on the influence between individuals such as families, friends or colleagues [13,14]. Sociological studies have shown that information coming from close relatives has strong impact on an individual’s decision and in this way a trend, behaviour or information can propagate on the network [1]. Consequently, finding the most influential users in a network becomes a critical task. The challenge of finding the *k* most influential users in a social network is called influence maximization [3]. The set containing these influential users is called a seed set. The activation of this set of nodes can maximize the expected spread of information in the network.

Influence maximization is widely studied and many different approaches have been considered to identify scalable solutions for finding the most influential users [15–19]. Many of these approaches did not consider the time required to achieve influence spread during the maximization process. But, in some cases, time could be a feature and has various levels of importance in different marketing applications. For instance, marketers in a clothing company have to advertise and maximize the number of sales of the winter products in cold seasons. As an example, customers may start to become aware of winter collection of boots and coats from early winter and there is a limited time for advertisement of such products. In addition to influence, i.e., the number of active users, diffusion time also becomes critical. In other applications, time may be less important, therefore, maximizing the influence over unlimited time would suffice.

Diffusion time is the elapsed time from the beginning of information spread until no other user can be activated. This time depends on the delay (time taken) by each user to spread information. Since users have different propagation delays, different seed sets may reach their final influence spread after spanning different diffusion times. Liu et al. [17] showed that a seed set which can maximize the amount of influence without considering any time bound is not necessarily the best choice for those applications which have a specific time bound.

Given that increasing influence and decreasing diffusion time are two adverse objectives, there would be several seed sets each having the best influence for a specific time bound. Finding all these sets at the same time will help decision makers to select the best seed set that fits to their needs.

The objective of this paper is to find near-optimal seed sets for different time bounds simultaneously. We have sought solutions that have more influence in specific time bounds and have compared our results with other approaches.

The major contributions of our paper are as follows:

- We show the relationship between influence and time in information diffusion process and propose a novel problem which aims to find near-optimal seed sets for different time bounds simultaneously.
- We propose an algorithm which is able to select the optimal seed sets from the results of NSGA-II algorithm [20].
- We propose two heuristics to estimate the amount of influence spread and diffusion time in the

NSGA-II algorithm in order to reduce the running time of the algorithm.

- We compare the results of our approach with an existing algorithm for influence maximization which considers time limitation in its calculations and also with two other heuristic methods, on three real datasets.

This paper is organized as follows. In section 2, we briefly review research in the area of influence maximization problem and explain multi-objective optimization and the NSGA-II algorithm. In section 3, we present our proposed method and explain how we have employed multi-objective solution for our problem and how we have handled the large scale issue of social networks. In section 4, we show the result of applying our approach on real datasets. Conclusion of our work and future work are brought in section 5.

2. Related works

The influence maximization problem aims to find *k* nodes in a social network that directly and indirectly influence the largest number of nodes under a predefined diffusion model. Richardson et al. [21,22] studied the problem of influence maximization and modelled it using the Markov random field method. This problem was first formulated as a discrete optimization problem by Kempe et al. in 2003 [16]. They modelled a social network as a directed graph, in which nodes are individuals and edges represent relationships between them. In their work, they presented two stochastic diffusion models: an independent cascade (IC) model and a linear threshold (LT) model. These have become baselines for other works on the influence maximization problem. In addition, they proved that the influence maximization problem is NP-hard and proposed a greedy algorithm with a (1-1/e) approximation guarantee of an optimal solution.

It is proved that the problem of calculating the exact influence spread of a seed set is NP-hard under IC [23] and LT [24] models. Due to the complexity of the simulation approach to find influence spread, a set of algorithms such as CELF [25], CELF++ [26], STATICGREEDY [27], RIS [28] and TIM [29] are proposed that improve the runtime of Kempe's algorithm, while maintaining the influence spread guarantee.

In contrast to approximation algorithms, several studies use heuristic methods for selection of influential nodes in influence maximization problem, among which, we can mention LDAG [24] and SIMPATH [30] for LT model as well as MIA [23,31] and IRIE [32] for IC model.

Most of the previous works in this area have focused only on the coverage of seed set. Tang et al. [33] introduced node diversity as a second objective function. They defined diversified social influence maximization, formulated it as an optimization problem and provided a greedy solution. Gunasekara et al. [20] used the NSGA-II algorithm to identify the set of key players in the social networks, by considering two objectives, eigenvector centrality and the distance between key players.

Very recently, some works have noticed to time aspect in the diffusion process. Gomez-rodriguez et al. [34,35] proposed methods for learning the network structure and transmission rates from temporal propagation data. Goyal et al. [36], suggested methods for estimating the parameters of static and time-dependent models from the propagation data. Mohammadi et al. [37], introduced Time-Sensitive Influence Maximization problem, in which the dependency of information value to time is incorporated into influence maximization process. They have mentioned that in the time-sensitive applications, not only the coverage size, but also the time of activation matters. Liu et al. [17] introduced time constrained influence maximization in which the goal is to find a set of users which can maximize the amount of influence in a limited time bound. They defined an influence spreading path concept and proposed scalable algorithms that can be scaled to large social networks. However in their work, one needs to run such an algorithm with different time bounds in order to find suitable seed sets for each particular time bound. The selection of seed nodes in the Liu's approach is in a greedy manner. A node will be added to the current seed set if it gives the highest influence gain in a limited time bound. This process is repeated until the desired number of seeds is achieved.

In our work, we have considered both the coverage size and the diffusion time as the objectives of influence propagation. In fact, increasing coverage size and decreasing diffusion time are two conflicting objectives that we consider in the selection of seed nodes. This introduce a new multi-objective optimization problem, which we employ NSGA-II algorithm for solving it. Given the massive scale of social networks, we propose two efficient methods for estimating influence and diffusion time to reduce the execution time of NSGA-II in our problem.

Furthermore, we discuss the results under time constraints and propose an algorithm for selecting influential nodes in specific time bounds. We compare the influence spread of our approach with results obtained by Liu et al. [17], for specific time bounds. The details of experiments are given in section 4.

2.1. Influence maximization and diffusion models

In this section, we present the influence maximization problem and the Independent Cascade (IC) model. We expand them to consider time in the diffusion process and present a latency aware IC model and a time constrained influence maximization problem.

Definition 1: Given a directed social network $G=(V, E)$, where V represents the set of nodes and E represents the set of edges between them, positive number $k < |V|$ and propagation probability P_{uv} between each edge uv, the goal of the influence maximization problem is to find the seed set S containing k nodes where the expected number of active nodes using S, $\sigma_T(S)$, is maximized at the end of the diffusion process.

Several models have been presented to demonstrate how influence propagates in a social network [16,38,39]. One of the most popular models for describing influence propagation is the independent cascade (IC) model which is presented by Kempe et al. [16]. In this model, each node can be in active or inactive state. Inactive nodes can become active but active nodes will stay active. In a diffusion process based on this model, each active node u which is activated at step t, has a single chance to activate all of its currently inactive neighbors at step t+1. An active node u can activate its inactive neighbor v with probability $P_{uv} \in [0, 1]$ which is called propagation probability and is assigned on the edge that connects them.

The IC model does not consider time in the diffusion process so it is not suitable to use this model when time needs to be considered. Liu et al. [17] expanded the IC model to consider time and presented a Latency Aware Independent Cascade (LAIC) model. In the LAIC model, if node u which is activated at time t, could successfully activate its inactive neighbor v, the activation time of node v will be $t+\delta_t$ with a probability $P_{uv}.P_u^{lat}(\delta_t)$ where δ_t is influencing delay and is drawn from a delay distribution $P_u^{lat}(\delta_t)$. Based on this model, they introduced the time constrained influence maximization problem.

Definition 2: Given a social network $G = (V, E)$, time bound T, positive integer $k < |V|$, activating probability $P_{uv} \in [0,1]$ and latency distribution P_u^{lat} for each $u \in V$, find a seed set $S \subset V$ of k nodes, such that the expected number of nodes influenced by S within time T, $\sigma_T(S)$, is maximized under the LAIC model [17].

2.2. Multi-objective optimization

Definition 3: The process of maximizing or minimizing multiple objective functions subject to a set of constraints is called multi-objective optimization. Multi-objective optimization can be mathematically formulated as Eq. (1):

$$\begin{aligned}&\max\left(f_1(x), f_2(x), \ldots, f_d(x)\right)\\&\textit{subject to}\ \ c_i(x) \leq 0\ ; i = 1,2,\ldots,m \qquad (1)\\&c_j(x) = 0\ ; j = 1,2,\ldots,e\end{aligned}$$

In this equation, the goal is to maximize *d* objective functions subject to *m* inequality constraints and *e* equality constraints. A similar equation can be written for the minimization problem such that minimizing *f(x)* is equal to maximizing *1/(f(x))* or *–f(x)*.

Since there are more than one objective functions in the multi-objective optimization problem, there does not exist a single solution that simultaneously optimizes all objectives. Instead, there exists a set of solutions called Pareto optimal or non-dominated sets which mean that no other solution dominates them [40]. Solution *A* dominates solution *B* if and only if *A* is better than *B* in all objectives. The set of all Pareto optimal solutions is called Pareto front. Solutions in Pareto front are tradeoffs between different objectives. Recently, multi-objective optimization methods are used for solving different problems in social networks. Gunasekara et al. [20] applied multi-objective optimization to find set of key players in a network that maximize both the average Eigenvector centrality and distance between the key players. In another work, Bucur et al. proposed a multi-objective evolutionary algorithm to maximize the influence and minimize the size of seed set concurrently [41]. Zuo et al. [42] used evolutionary multi-objective optimization for personalized recommendation and in [5,7] multi-objective evolutionary algorithms are proposed for community detection problem.

There are many different algorithms for solving the multi-objective optimization problem. These include PAES [43], SPEA [44], and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [45]. The NSGA-II is the most powerful algorithm among these because of its lower computational complexity and its ability to provide an approach which is able to come closer to the true Pareto front than other methods.

Considering the two objectives of decreasing diffusion time and increasing influence, we used the NSGA-II algorithm to find the Pareto front which contains seed sets that are non-dominant to each other in terms of diffusion time and influence. By yielding all of the potentially optimal solutions, a decision maker can make tradeoffs between the two objectives. In the next section, we briefly describe the NSGA-II algorithm.

2.3. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm is a multi-objective evolutionary algorithm, which was first presented by Deb et al. [45]. In this algorithm, a random parent population is initially created. NSGA-II then uses a fast non-dominated sorting approach to find different fronts from a population. Each front contains individuals that are non-dominated to each other, which means that no individual yields better results for all the objective functions than other individuals in the same front with respect to all objectives.

Each individual will be assigned a fitness (rank) equal to its nondomination level (1 is the best, 2 is next best level and so on). Binary tournament, one point crossover and mutation operators will be used to create an offspring population and after combining them with a parent population a mating pool will be created. The fast non-dominated sorting approach will be applied to this combined population and a fitness value will be assigned to each individual.

New generation will be achieved using a non-domination concept, which selects a new population from individuals in lower fronts. If a population size is greater than the number of individuals in the first front, all its members will be added to the new population. This process will continue for all following fronts until a predefined number of individuals, i.e. population size, have been reached. During the selection process, if the number of individuals in the next front is greater than the remaining free space in the new population, NSGA-II uses a crowded-comparison approach. For employing this approach, in addition to fitness value a new parameter called crowding distance is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbors.

Having the crowding distance and the fitness (rank) of individuals, the selection process works as follow. Between two solutions with differing nondomination ranks, the selection process prefers the solution with the lower (better) rank. If both solutions belong to the same front, then it prefers the solution that is located in a lesser crowded region. This approach guides the selection process toward a uniformly spread-out Pareto optimal front [45]. This procedure will be continued until a predefined condition such as the maximum number of iterations is satisfied.

3. Methods

In this section, we explain our proposed method to find influential users for different time bounds. First, we show the relationship between the amount of influence and the diffusion time of different seed sets. Then we apply the NSGA-II algorithm to find the seed sets in Pareto front which are non-dominant to each other while they dominate other solutions that are not present in the Pareto front. Since diffusion in the LAIC model is a stochastic process, the expected influence of each chromosome in NSGA-II algorithm should be calculated using Monte Carlo simulation [17] which is computationally expensive. To reduce time complexity, instead of using Monte Carlo simulation, we use two heuristics to calculate amount of influence and diffusion time of each seed. We show that all solutions found in Pareto optimal are not necessarily acceptable for a time bound equal to their diffusion time. We propose an algorithm to select the best seed set from the Pareto front for a given time bound. This is explained in the following sections.

3.1. Relation between diffusion time and influence

In the real world social networks, an active node can activate its inactive neighbours with a delay. It is trivial to get more active nodes at the cost of increasing the diffusion time. This property of influence spread is shown in Fig 1 where influence spread is measured as a function of time. In this figure we used a Dolphin social network [46], which is a social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand. We considered all possible combinations of seed sets of size 2. We used Poisson distribution to estimate delay distribution as its parameter chosen uniformly at random between 1 and 20 for each node. We used a weighted cascade approach to assign propagation probability P_{uv} to the edge connecting nodes *u* and *v*, which is equal to 1/(in-degree of node *v*) [23]. Monte Carlo (MC) simulation with 5000 iterations is used to calculate the influence of each node. After calculating the influence and diffusion time of each seed set node, we considered the maximum activation time among nodes as diffusion time.

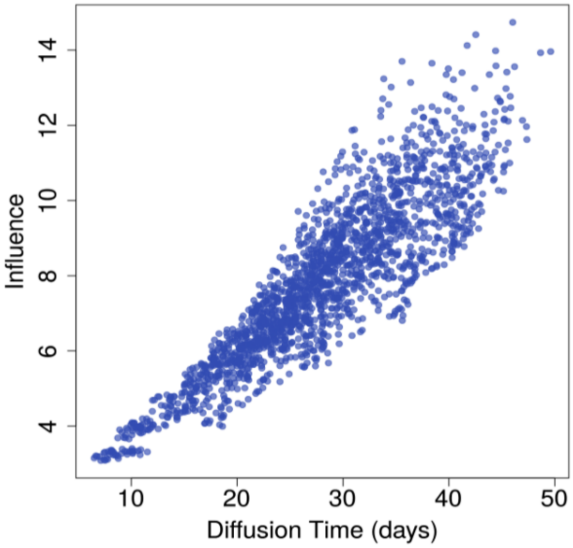

Fig 1. Influence spread vs. diffusion time in Dolphin social network (*k*=2)

In some applications, diffusion time is of paramount importance because influence on users after a specific amount of time becomes worthless [17,47] or less valuable [37], although, in some other applications time may not be so critical. Consequently, finding the optimal seed set for a particular time bound might be required for an application of interest.

As discussed earlier, influence and diffusion time are two adverse objectives (higher influence is desired, but in a shorter time), these objectives lead us to use multi-objective optimization to find optimal solutions for different time bounds.

3.2. Applying NSGA-II algorithm to influence maximization problem

We use the NSGA-II algorithm to find the Pareto optimal set of size 2 for the Dolphin social network. We consider 300 iterations and 100 random binary chromosomes as the initial population. Each chromosome represents a seed set. The size of each chromosome is equal to the number of nodes in the network. If a node is included in the seed set, we put 1 in its position in the chromosome and otherwise we place 0. We also used one point crossover and bitwise mutation to create offspring. After applying these operators to parents, each new offspring is checked to ensure that the number of seeds in a chromosome is equal to the desired size of the seed set. Fig 2 shows the result of applying the NSGA-II algorithm to the Dolphin social network.

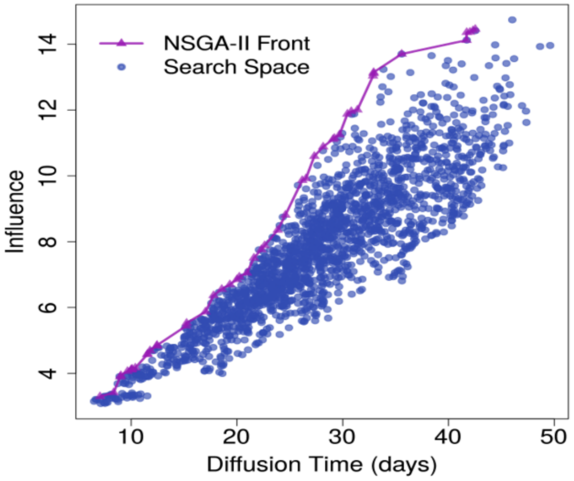

Fig 2. *Pareto optimal calculated with NSGA-II in Dolphin social network (k=2)*

The triangles in Fig 2 show the resulting Pareto front of using the NSGA-II algorithm. These sets are non-dominant to each other, which means that none of them is better than the other in both influence and diffusion time.

The process of optimizing objectives, i.e., diffusion time and influence, is achieved without considering any time bound and we do not limit any of the seed sets to stop spreading their influence after a time bound has passed. As a result, a seed set with high influence and high diffusion time, may also be the best choice for smaller time bounds, which means that if we calculate the influence of this seed set, considering time limitation, we can earn more influence than the influence earned by a seed set which Pareto front has chosen for that time value. We calculate the influence of a seed set with limited time bound using Liu's approach [17].

An example of calculating the influence of seed sets belonging to the optimal front with higher diffusion time by imposing a lower time bound is shown in Fig 3. Considering the two seed sets shown in triangles, i.e., seed set s_j with influence $\sigma_{\mathrm{sj}} = 10.5$ and diffusion time $T_j = 36$ and seed set s_i with influence $\sigma_{\mathrm{si}} = 4.1$ and time $T_i = 10$. If we calculate the influence of seed set s_j with time bound of T_i (i.e. T=10), the amount of earned influence is equal to 7.3, (crossed circle in Fig 3) which is higher than influence of seed set s_i that is in the Pareto optimal set.

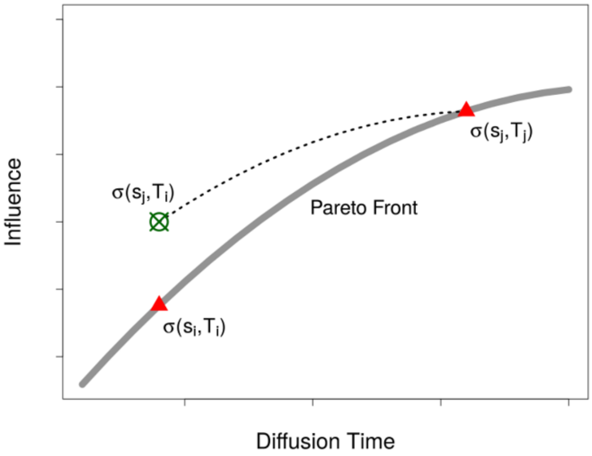

Fig 3. Recalculating the influence of seed sets in the optimal front by imposing a lower time bound

Based on this observation, we compute the influence of seed sets in the Pareto front with higher diffusion time by imposing lower time bounds, to not miss better seed sets for each time bound. Therefore, we present our Selecting Optimal Set (SOS) algorithm which is able to assign best seed sets to each specific time bound among seed sets found in Pareto optimal.

Using Algorithm 1 (SOS algorithm), we are able to find the best seed sets with highest influence for different time bounds. The procedure of this algorithm is described in the next subsection.

3.3. Selecting Optimal Set algorithm

We propose the SOS algorithm to find the best seed sets with the highest influence for different time bounds. The SOS algorithm is shown in Algorithm 1. Line 2 in the algorithm sorts the Pareto optimal sets based on time in increasing order, which allows us to compare seed sets according to their influence and diffusion time. Line 3 to 10, compare the influence of seed set s_i with the influence of other seed sets that are calculated in a limited time bound equal to the diffusion time of seed set s_i (T_i). Therefore, if seed set s_j $(j \neq i)$ has the most influence for time bound T_i among all other seed sets, it will be considered in the final result as the best seed for time bound equal to T_i. As in the procedure of Algorithm 1, we will eliminate some seed sets that are dominated by others.

Algorithm 1: SOS

```
Input: network $G=(V,E)$, Pareto optimal set $S$ of seed sets,
Output: New set of seed sets ($N$)
$N = \varnothing$
Sort $S$ based on time, in increasing order
for all seed set $s_i \in S$ do
    $N(i) = s_i$
    for all seed set $s_j \in S$ and $j > i$ do
        calculate influence of seed set $s_j$ with
        time bound equal to diffusion time of seed set $s_i$ ( $\sigma_{new}(S_j, T_i)$ )
        if ( $\sigma_{new}(S_j, T_i) > \sigma(N(i))$ ) then
            $N(i) = s_j$
        end if
    end for
end for
return $N$
```

If a special time bound is desired which is not provided in the output Pareto front, we compute the influence of all the seed sets in the Pareto front that have diffusion times higher than that particular time bound, using a limited time bound equal to that time, and report the best influence at that time bound.

3.4. Efficient Computation of Influence Spread and Diffusion Time

Although running a Monte Carlo simulation returns a good estimation of influence spread and diffusion time for a seed set, it demands huge computational time. In our proposed method, we have to use several chromosomes as seed sets, and need to find the influence and diffusion time of all of them in each iteration. Therefore, employing Monte Carlo simulations would be impractical when dealing with large social networks. To resolve this difficulty and reduce the computational requirement, we use two heuristics to calculate the influence and diffusion time of each seed set as described in the following sections.

3.4.1. Estimating Expected Influence

We use the Expected Diffusion Value (EDV) method proposed by Jiang et al. [48] to directly calculate the expectation of influence for a seed set without performing Monte Carlo simulation. Suppose N_{out} is the set of out-neighbours of seed set S, $N_{in}(w)$ is the set of in-neighbours of node $w \in N_{out}$ that belongs to seed set S, and P_{vw} is the probability that node v activates node w. An estimation of influence of seed set S can be calculated using Eq. (2)

$$inf_S = \sum_{w \in N_{out}} (1 - \prod_{v \in Nin(w)} (1 - P_{vw})) \tag{2}$$

In the above equation, $\prod_{\mathrm{v} \in \mathrm{Nin}(\mathrm{w})} (1 - \mathrm{P}_{\mathrm{vw}})$ shows the probability that node w is not activated by any of its neighbours that belong to S. Accordingly, $(1 - \prod_{\mathrm{v} \in \mathrm{Nin}(\mathrm{w})} (1 - \mathrm{P}_{\mathrm{vw}}))$ shows the probability that w is influenced by at least one of seed nodes. The sum of these values over all $\mathrm{w} \in \mathrm{N}_{\mathrm{out}}$ can be considered as an estimation of diffusion value of S.

3.4.2. Estimating Expected Diffusion Time

To reduce the cost of calculating diffusion time, we propose a method to compute the average diffusion time of a seed set. Consider a graph of three nodes as shown in Fig 4. In this figure, nodes u and v are seed sets which try to activate node w with probability P_{uw} and P_{vw}, respectively. The parameters of the delay distributions (Poisson distribution) for these two nodes are λ_v and λ_u .

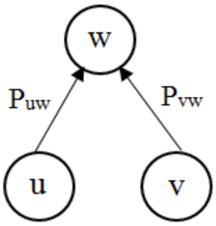

Fig 4 A graph containing three nodes where u and v are seed sets.

There are four different scenarios for activation of node w by u and v:

1. Node u successfully activates node w, but v fails to do so. The probability of occurrence of this is $P_{uw} . (1-P_{vw})$.

2. Node *v* successfully activates node *w*, but *u* fails to do so. The probability of occurrence of this is *(1-P_{uw}).P_{vw}*.
3. Both *u* and *v* successfully activate *w*. The probability of occurrence of this is $P_{uw}.P_{vw}$.
4. Both *u* and *v* fail to activate *w*, therefore *w* stays inactive. The probability of occurrence of this is *(1-P_{uw}).(1-P_{vw})*.

Based on these scenarios we can calculate the average activation time of node *w*, i. e. $\mathrm{E}\left(\mathrm{T}_{\mathrm{w}}\right)$ using Eq. (3):

$$E\left(T_w\right)=\frac{P_{uw}\left(1-P_{vw}\right)\delta\left(\lambda_u\right)+P_{vw}\left(1-P_{uw}\right)\delta\left(\lambda_v\right)}{1-\left(1-P_{uw}\right)\left(1-P_{vw}\right)}+\frac{P_{uw}P_{vw}E\left(\min\left(\delta\left(\lambda_u\right),\delta\left(\lambda_v\right)\right)\right)}{1-\left(1-P_{uw}\right)\left(1-P_{vw}\right)} \tag{3}$$

Where $\delta\left(\lambda\right)$ is the activation delay with Poisson distribution and mean parameter λ. The expectation of such delay from Poisson distribution is $E[(\delta\left(\lambda\right))]=\lambda$. We calculate the expectation of minimum of two Poisson random numbers using Eq. (4):

$$\begin{aligned}
&E\left(\min\left(\delta(\lambda_u),\delta\left(\lambda_v\right)\right)\right)=\sum_{x=0}^{\infty}xP\left(\min\left(\delta\left(\lambda_u\right),\delta\left(\lambda_v\right)\right)=x\right)\\
&=\sum_{x=0}^{\infty}\left(\sum_{j=1}^{x}1\right)P(\min\left(\delta\left(\lambda_u\right),\delta\left(\lambda_v\right)\right)=x\right)=\sum_{j=0}^{\infty}\sum_{x=j+1}^{\infty}P(\min\left(\delta\left(\lambda_u\right),\delta\left(\lambda_v\right)\right)=x\right)\\
&=\sum_{x=0}^{\infty}P(\min\left(\delta(\lambda_u),\delta\left(\lambda_v\right)\right)>x\right)=\sum_{x=0}^{\infty}P\left(\delta\left(\lambda_u\right)>x\right)P(\delta\left(\lambda_v\right)>x)\\
&=\sum_{x=0}^{\infty}\sum_{i=x}^{\infty}\frac{e^{-\lambda_u}\lambda_u^{\,i}}{i!}\sum_{j=x}^{\infty}\frac{e^{-\lambda_v}\lambda_v^{\,j}}{j!}
\end{aligned} \tag{4}$$

We use the above equation to estimate the average diffusion time of a seed set. In order to avoid premature convergence, we extend these calculations to two neighbouring hops when more than 35% of chromosomes become similar.

In the next section we show the experimental results using real world social network datasets.

4. Results and Discussion

To evaluate our method, we applied it on three real world networks. The properties of these networks are shown in Table 1.

Table 1 Dataset Properties

Network	WikiVote	NetHEPT	Epinions
Number of Nodes	7115	15K	75K
Number of Edges	103K	62K	508K
Clustering Coefficient	0.2089	0.3120	0.1378

The first dataset, WikiVote, is a Wikipedia voting network, in which nodes represent users and edges from node *i* to *j* represents that node *i* votes for node *j*. This dataset can be obtained from [49].

NetHEPT, is a collaboration network extracted from arXiv.org's High Energy Physics Theory section. Nodes in this network represent authors and if two authors collaborate, there is an arc in both directions between them. This dataset is obtainable from [50].

The last dataset, Epinions, is related to a who trust-whom social network of a general consumer review site Epinions.com. Each node in this dataset shows a member of the site and a directed edge from *u* to *v* means *v* trusting *u* (and thus *u* has influence on *v*). This dataset can be obtained from [49].

For our experiments, we use binary chromosomes, with one point crossover and bitwise mutations. The crossover probability and mutation probability are set to 0.9 and 0.01, respectively. We decrease mutation probability in each iteration by 5%.

We consider more chromosomes for larger networks, since the search space is larger in those graphs. It should be mentioned that since we represent a candidate solution as a bit string of size equal to the number of nodes in the target network, our method consumes high memory, which makes it hardly scalable on large networks. Using a more memory-efficient individual representation is a promising direction and is left as our future work. The number of chromosomes and iterations we used for different datasets are shown in Table 2.

Table 2. *Number of iterations and chromosomes in employed datasets*

Dataset	WikiVote	NetHEPT	Epinions
Number of chromosomes	1000	2000	3000
Number of iterations	2000	3000	4000

Poisson distribution is used to create delays in activating each node and its parameters are drawn randomly between 1 and 20. We pre-calculate the values of the expected minimum of each 2 Poisson numbers using Eq. (4) to avoid recalculating them in each iteration during optimization. We compare our results with the Influence Spreading Path (ISP) method proposed by Liu et al. [17]. In this method they calculate $\sigma_T(S \cup \{v\})$ and $\sigma_T(S)$, ($\sigma_T(S)$ is the influence of seed set *S* in time bound *T*) by using an influence spreading path and using a greedy algorithm to select the optimal seed set. The threshold parameter for this algorithm is set to 10^{-5}. This parameter controls the number of influence spreading paths for the ISP method [17]. The parameter for the Poisson distribution of each node is a random number between 1 and 20. In addition, the results are compared with Degree heuristic and Random method. In Degree heuristic, *k* nodes with highest degree are selected as seed set. In Random method, *k* nodes are selected randomly.

Fig 5 to 7 show the results of influence spread for WikiVote, NetHEPT, and Epinions datasets, respectively, with different time bounds for seed sets of size 50. In these figures, the horizontal axis represents different time bounds and the vertical axis represents the influence spread of selected seed set by each method. As shown in Fig 5 to 7, our approach outperforms Degree and Random methods in all cases. Furthermore, it is able to find seed sets that have equal or higher influence than ISP method. Consequently, our approach not only finds influential nodes for different time bounds simultaneously but also for a specific time bound it outperforms other methods in terms of influence spread.

As it is illustrated in Fig 5, in WikiVote dataset, the influence spread of different methods after time bound 70 are not significantly changed. This indicates that most of the influenced nodes in WikiVote dataset can be activated before *T*= 70. In NetHEPT dataset, the influence spread is not changed significantly after *T*=85 but the slope of increasing influence spread is steeper in NetHEPT in comparison to WikiVote dataset, since NetHEPT is a larger network and diffusion of influence in this network takes more time. As it is observable in Fig 7, Epinions has the steepest slope of diffusion value over time in comparison to other datasets.

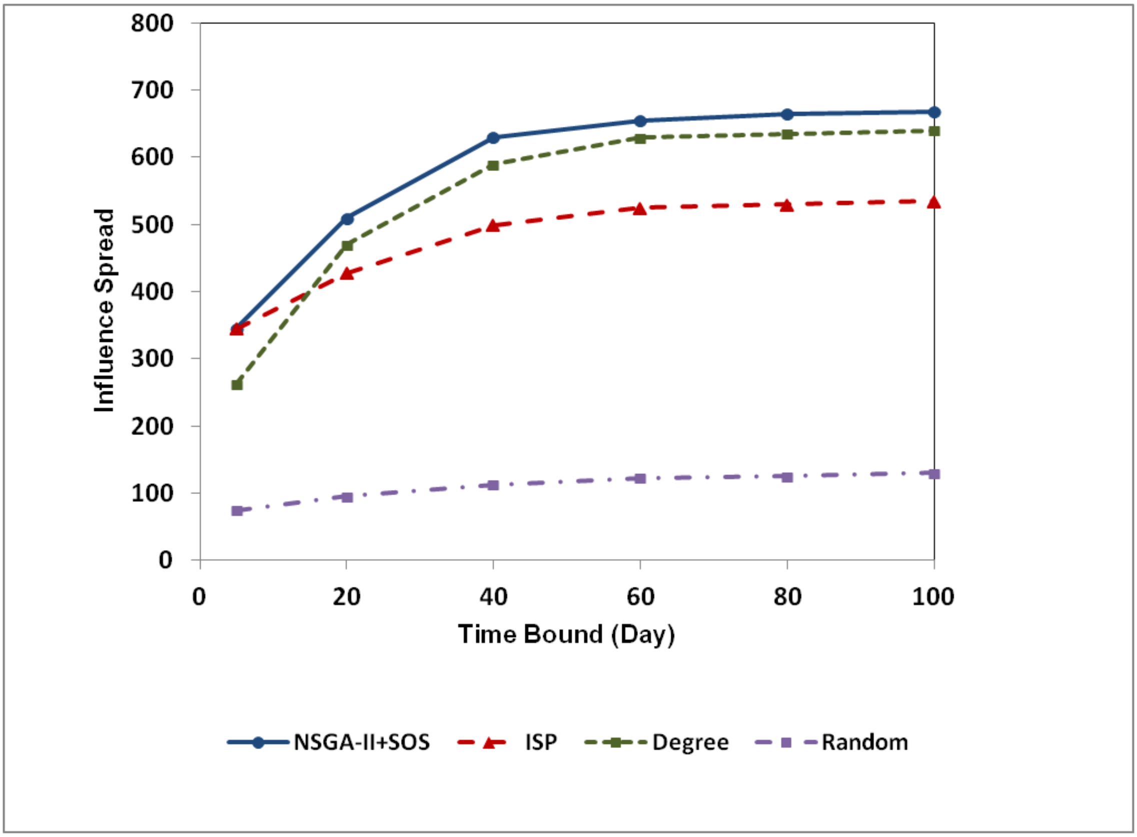

Fig 5 Results of influence spread on WikiVote dataset (K=50)

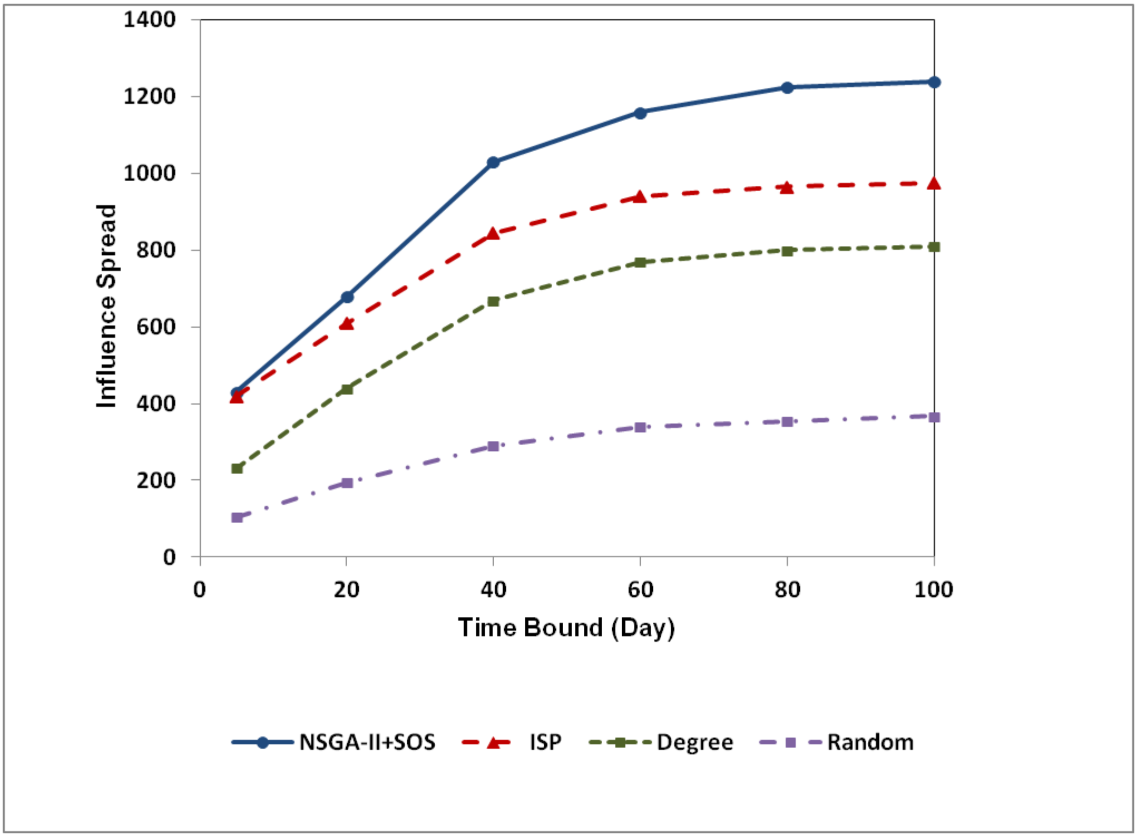

Fig *6 Results of influence spread on NetHEPT dataset (*K=*50)*

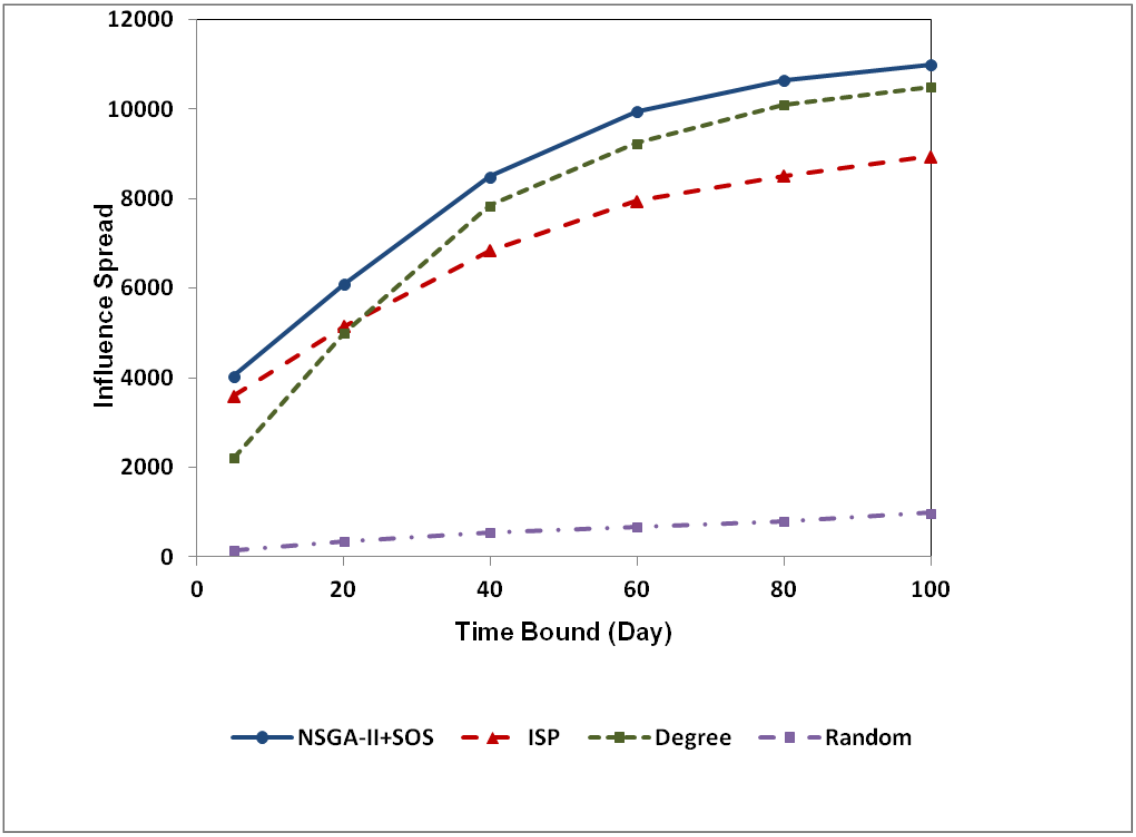

Fig 7 *Results of influence spread on Epinions dataset (*K=*50)*

5. Conclusions

In this paper, we have considered both the coverage size and diffusion time as the objectives of optimization in diffusion process. Considering two conflicting objectives, increasing influence and decreasing diffusion time, leads to a new multi-objective optimization problem which is neglected in previous works. For solving this problem, we employed NSGA-II algorithm which finds a set of non-dominant seed sets in terms of diffusion time and influence. Providing the set of all potentially optimal solutions, help a decision maker evaluate the tradeoffs between the two objectives. Furthermore, we proposed an algorithm to select best seed set for different time bounds from the results found by the NSGA-II algorithm. We used two heuristics to calculate the amount of diffusion time and influence of each seed set to reduce the computational time.

Our approach was applied to three real world datasets, which resulted in a significant increase spread over the state of the art algorithm. In fact, our proposed method not only can find influential nodes for different time bounds simultaneously but also for a specific time bound it outperforms previous works in terms of influence spread.

Although we have addressed a novel problem in this paper, there is space to improve the method in the future. One important aspect in applying multi-objective algorithms is the memory usage. Therefore, designing a memory-efficient algorithm can be considered as a future work. Reducing the memory usage will increase the scalability of our method. In addition, the proposed method in this paper is studied under LAIC model which is an extension of IC model. Extending other diffusion models such as Linear Threshold model to delayed version and analysing our algorithm under them is a promising research direction.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors

References

[1] J. Sun, J. Tang, A Survey of Models and Algorithms for Social Influence Analysis, in: C.C. Aggarwal (Ed.), Soc. Netw. Data Anal., 1st ed., Springer, 2011: pp. 177–214.

[2] W. Chen, L.V.S. Lakshmanan, C. Castillo, Information and Influence Propagation in Social Networks, First ed, Morgan & Claypool, Cleveland, 2013.

[3] H. Li, J.-T. Cui, J.-F. Ma, Social Influence Study in Online Networks: A Three-Level Review, J. Comput. Sci. Technol. 30 (2015) 184–199.

[4] R. Shang, S. Luo, Y. Li, L. Jiao, R. Stolkin, Large-scale community detection based on node membership grade and sub-communities integration, Phys. A Stat. Mech. Its Appl. 428 (2015) 279–294.

[5] R. Shang, S. Luo, W. Zhang, R. Stolkin, L. Jiao, A multiobjective evolutionary algorithm to find community structures based on affinity propagation, Phys. A Stat. Mech. Its Appl. 453 (2016) 203–227.

[6] R. Shang, W. Zhang, L. Jiao, R. Stolkin, Y. Xue, A community integration strategy based on an improved modularity density increment for large-scale networks, Phys. A Stat. Mech. Its Appl. 469 (2017) 471–485.

[7] R. Shang, H. Liu, L. Jiao, Multi-objective clustering technique based on k-nodes update policy and similarity matrix for mining communities in social networks, Phys. A Stat. Mech. Its Appl. 486 (2017) 1–24.

[8] G. Nandi, A. Das, A Survey on Using Data Mining Techniques for Online Social Network Analysis, Int. J. Comput. Sci. 10 (2013) 162–167.

[9] Y. Yang, R.N. Lichtenwalter, N. V. Chawla, Evaluating link prediction methods, Knowl. Inf. Syst. 45 (2015) 751–782.

[10] J.X. Yang, X.D. Zhang, Revealing how network structure affects accuracy of link prediction, Eur. Phys. J. B. 90 (2017) 157–164.

[11] F. Benevenuto, T. Rodrigues, M. Cha, V. Almeida, Characterizing user behavior in online social networks, in: Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf., Chicago, USA, 2009: pp. 49–62.

[12] A. Squicciarini, S. Rajtmajer, C. Griffin, Positive and negative behavioral analysis in social networks, Data Min. Knowl. Discov. 7 (2017) 1–12.

[13] C. Long, R.C.-W. Wong, Viral marketing for dedicated customers, Inf. Syst. 46 (2014) 1–23.

[14] B.-L. Zhang, Z.-Z. Qian, W.-Z. Li, B. Tang, S.-L. Lu, X. Fu, Budget Allocation for Maximizing Viral Advertising in Social Networks, J. Comput. Sci. Technol. 31 (2016) 759–775.

[15] N. Barbieri, F. Bonchi, Influence Maximization with Viral Product Design, in: Proc. 2014 SIAM Int. Conf. Data Min., Philadelphia, USA, 2014: pp. 55–63.

[16] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the Spread of Influence through a Social Network, in: Proc. Ninth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD ’03, Washington DC, USA, 2003: pp. 137–146.

[17] B. Liu, G. Cong, Y. Zeng, D. Xu, Y.M. Chee, Influence Spreading Path and Its Application to the Time Constrained Social Influence Maximization Problem and Beyond, IEEE Trans. Knowl. Data Eng. 26 (2014) 1904–1917.

[18] N. Ohsaka, T. Akiba, Y. Yoshida, K. Kawarabayashi, Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations, in: Proc. Twenty-Eighth AAAI Conf. Artif. Intell., Québec City, Canada, 2014: pp. 138–144.

[19] I. Roelens, P. Baecke, D.F. Benoit, Identifying influencers in a social network: The value of real referral data, Decis. Support Syst. 91 (2016) 25–36.

[20] R.C. Gunasekara, K. Mehrotra, C.K. Mohan, Multi-Objective Optimization to Identify Key Players in Social Networks, Soc. Netw. Anal. Min. 5 (2015) 443–450.

[21] P. Domingos, M. Richardson, Mining the Network Value of Customers, in: Proc. Seventh ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., San Francisco, USA, 2001: pp. 57–66.

[22] M. Richardson, P. Domingos, Mining Knowledge-sharing Sites for Viral Marketing, in: Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., Edmonton, Canada, 2002: pp. 61–70.

[23] C. Wang, W. Chen, Y. Wang, Scalable influence maximization for independent cascade model in large-scale social networks, Data Min. Knowl. Discov. 25 (2012) 545–576.

[24] W. Chen, Y. Yuan, L. Zhang, Scalable Influence Maximization in Social Networks under the Linear Threshold Model, in: Proc. 2010 IEEE Int. Conf. Data Min., Sydney, Australia, 2010: pp. 88–97.

[25] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective Outbreak Detection in Networks, in: Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD ’07, San Jose, USA, 2007: pp. 420–429.

[26] A. Goyal, W. Lu, CELF++: Optimizing the Greedy Algorithm for Influence Maximization in Social Networks, in: Proc. 20th Int. Conf. Companion World Wide Web, Hyderabad, India, 2011: pp. 47–48.

[27] S. Cheng, H. Shen, J. Huang, G. Zhang, X. Cheng, StaticGreedy : Solving the Scalability-Accuracy Dilemma in Influence Maximization, in: Proc. 22nd ACM Int. Conf. Inf. Knowl. Manag., San Francisco, USA, 2013: pp. 509–518.

[28] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing Social Influence in Nearly Optimal Time, in: Proc. 25nd Annu. ACM-SIAM Symp. Discret. Algorithm, Portland, Oregon, USA, 2014: pp. 946–957.

[29] Y. Tang, X. Xiao, Y. Shi, Influence Maximization : Near-Optimal Time Complexity Meets Practical Efficiency, in: Proc. 2014 ACM SIGMOD Int. Conf. Manag. Data, New York, NY, USA, 2014: pp. 75–86.

[30] A. Goyal, W. Lu, L.V.S. Lakshmanan, SIMPATH : An Efficient Algorithm for Influence Maximization under the Linear Threshold Model, in: Proc. 2011 IEEE Int. Confereance Data Min., Vancouver, Canada, 2011: pp. 211–220.

[31] W. Chen, C. Wang, Y. Wang, Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale

Social Networks, in: Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., Washington, DC, USA, 2010: pp. 1029–1038.

[32] K. Jung, W. Heo, W. Chen, IRIE: A Scalable Influence Maximization Algorithm for Independent Cascade Model and Its Extensions, in: Proc. 12th IEEE Int. Conf. Data Min., Brussels, Belgium, 2012: pp. 1–20.

[33] F. Tang, Q. Liu, H. Zhu, E. Chen, F. Zhu, Diversified social influence maximization, in: Proceeding Adv. Soc. Networks Anal. Min., Beijing, China, 2014: pp. 455–459.

[34] M. Gomez-rodriguez, D. Balduzzi, B. Schölkopf, Uncovering the Temporal Dynamics of Diffusion Networks, in: Proc. Twenty-Eighth Int. Conf. Mach. Learn., Bellevue, Washington, USA, 2011: pp. 561–568.

[35] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, B. Schölkopf, Influence Estimation and Maximization in Continuous-Time, ACM Trans. Inf. Syst. 34 (2016) 1–33.

[36] A. Goyal, F. Bonchi, L.V.S. Lakshmanan, Learning Influence Probabilities In Social Networks, in: Proc. Third ACM Int. Conf. Web Search Data Min., New York, NY, USA, 2010: pp. 241–250.

[37] A. Mohammadi, M. Saraee, A. Mirzaei, Time-sensitive influence maximization in social networks, J. Inf. Sci. 41 (2015) 765–778.

[38] M. Lahiri, M. Cebrian, The Genetic Algorithm as a General Diffusion Model for Social Networks, in: Proc. Twenty-Fourth AAAI Conf. Artif. Intell., Atlanta, USA, 2010: pp. 494–499.

[39] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, W. Wei, Y. Wang, Y. Yuan, Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate, in: Proc. 2011 SIAM Int. Conf. Data Min., Arizona, USA, 2011: pp. 379–390.

[40] A. Zhou, B. Qu, H. Li, S. Zhao, P. Nagaratnam, Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm Evol. Comput. 1 (2011) 32–49.

[41] D. Bucur, G. Iacca, A.M. B, G. Squillero, A. Tonda, Multi-objective Evolutionary Algorithms for Influence Maximization in Social Networks, in: Eur. Conf. Appl. Evol. Comput., Amsterdam, The Netherlands, 2017: pp. 221–233. doi:10.1007/978-3-319-55849-3.

[42] Y. Zuo, M. Gong, J. Zeng, L. Ma, L. Jiao, Personalized Recommendation Based on Evolutionary Multi-Objective Optimization, IEEE Comput. Intell. Mag. 10 (2015) 51–62.

[43] J. Knowles, D. Corne, The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation, in: Proc. 1999 Congr. Evol. Comput., Washington, D.C., USA, 1999: pp. 98–105.

[44] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications, Swiss Federal Institute of Technology, 1999.

[45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.

[46] Dolphin social network, (2015). https://networkdata.ics.uci.edu/data.php?id=6.

[47] B. Liu, G.C. Dong Xu, Y. Zeng, Time Constrained Influence Maximization in Social Networks, in: Proc. IEEE 12th Int. Conf. Data Min., Brussels, Belgium, 2012: pp. 439–448.

[48] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, K. Xie, Simulated Annealing Based Influence Maximization in Social Networks, in: Proc. Twenty-Fifth AAAI Conf. Artif. Intell., San Francisco, USA, 2011: pp. 127–132.

[49] J. Leskovec, A. Krevl, SNAP Datasets: Stanford Large Network Dataset Collection, (2014). http://snap.stanford.edu/data.

[50] Nethept Dataset, (2015). http://research.microsoft.com/en-us/people/weic/projects.aspx.