Swarm and Evolutionary Computation 51 (2019) 100614

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Identifying influential nodes based on ant colony optimization to maximize profit in social networks

Chiman Salavati, Alireza Abdollahpouri *

Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Keywords:
Influential node
Ant colony optimization
Viral marketing
Social network

ABSTRACT

One of the most important applications for identification of influential nodes in social networks is viral marketing. In viral marketing, there are valuable users from which companies or smaller businesses benefit most at the lowest cost. Inspired from the behavior of real ants and based on the ant colony optimization algorithm, we propose new methods named PMACO and IMOACO in this paper to find the most valuable users. First, the influence graph is derived from the analysis of users' interactions and communications in a social network. The negative influence among users is also considered in the process of generating the influence graph. For reduction of computational complexity and removal of unimportant nodes from the influence graph, the nodes the levels of influence of which on their neighbors are less than a specific threshold value are eliminated. Then, the representation of the search space as a weighted graph is constructed by the remaining nodes, where the weight of each edge is the similarity between the two nodes of which that edge is composed. Next, the ants begin their search processes with the goal of maximizing profit and minimizing the similarity among the selected nodes. Assessments have been made on real and synthetic datasets, and compared the proposed algorithm with well-known ones. The results of the experiments demonstrate the efficiency of the proposed algorithm.

1. Introduction

Social networks are changing the world in several ways, and are playing a fundamental role as a medium through the diffusion process [1]. Social networks are the best means of spreading information as compared to other social media such as newspapers and the television. A social network can be represented by a graph model where the nodes represent the users, and the edges indicate the interactions between them. In a social network, the users establish different communications and communities, and their opinions influence others users' purchase decisions. Viral marketing is the process of targeting a small subset of potential customers as a target (seed) set such that they can influence as many people in the network as possible [2–6]. Due to the large number of users in a social network, it becomes a critical task in viral marketing to find the most valuable users that obtain maximum profit at minimum cost [6,7].

There are many different approaches to evaluation of the importance of a node by identifying the number of users that are activated by that node [8–14]. Gateway Local Ranking (GLR) is a community-based algorithm in which a community detection approach is first applied to extract community structures of the network. After ignoring the relationship between communities, one best node as local critical node for each community is extracted according to a centrality measure. Then, with the consideration of interconnection links between communities, another best node as gateway node is found [13]. Degree Punishment (DP) [14] is a heuristic method which selects spreaders sequentially by applying a punishing strategy to the neighbors of already selected spreaders. The above-mentioned methods assume that the influences between users are positive, such that if a user is selected as an influential user, (s)he will have positive influence on others' purchase decisions. In the real world, however, there is also negative influence among users that spreads through social networks besides the positive influence, and the above-mentioned works have not considered this. In other words, if users' appropriate topological conditions in the social network graph are simply considered, users may be selected as target users who attempt to propagate negative intrusions over the network, and, therefore, have reverse outcomes in applications such as viral marketing. Therefore, negative influence should be taken into account in addition to positive influence.

Furthermore, companies have limited advertising budget to inform a large number of users; therefore, they select some people to spread the information over the network at the beginning, enabling them to

* Corresponding author.
E-mail addresses: chiman.salavati@eng.uok.ac.ir (C. Salavati), abdollahpour@uok.ac.ir (A. Abdollahpouri).

https://doi.org/10.1016/j.swevo.2019.100614
Received 6 May 2018; Received in revised form 30 October 2019; Accepted 30 October 2019
Available online 2 November 2019

convince people to use a product and spread information regarding that amongst their friends [15]. These companies pay a *cost* for each targeted user, for example by giving him/her free samples of a product, expecting that the selected users will obtain *profit* by recommending the product to their friends, and the diffusion process continues. The main objective of the company is to select the most valuable users by which maximum profit is achieved whereas total cost is minimized. For reduction of the costs of companies, it is important to select users that are dissimilar in terms of the users that are influenced by them. Therefore, it is a big challenge to find a feasible solution to obtain the set of most valuable people in the network.

The existing methods of finding the top-k influential nodes mainly focus on the greedy algorithm. Kempel et al. [2,16] formulated the top-k influence maximization problem as a combinational optimization problem. New enhancements based on the greedy algorithm have been proposed, e.g. the CELF algorithm [17] and the NewGreedy algorithm [18]. Although these enhancements greatly improve the efficiency of the greedy algorithm, it is still computationally challenging to use them. Furthermore, these methods ignore the relations among users, particularly negative influence and similarity. The influence maximization problem has been formally defined by Kempe et al. [16] and Domingos and Richardson [19]. Kempe et al. approached the problem from the perspective of the linear threshold (LT) and independent cascade (IC) diffusion models. Many of the existing approaches to solving the influence maximization problem are based on approximation algorithms, and assume that the objective function is monotonic and sub-modular [18, 20–22]. As identified by Borodin et al. [23], however, there is a complex, broad family of diffusion models, and the properties of monotonicity and sub-modularity may not hold, in which case the greedy approach cannot be used [24]. Swarm Intelligence includes a population of artificial agents that seek to simulate the behavior of a group of animals within the natural world. A simple task is performed by each agent alone, but they solve a complex problem together. One of the most popular algorithms in this area is ACO (Ant Colony Optimization), as it is flexible, is implemented easily, and enables global and local search. ACO has received extensive attention due to its successful application to many NP-hard combinatorial optimization problems [25].

In this paper, the influence of a node is investigated from another perspective, and new algorithms called *PMACO* and *IMOACO* are presented based on ACO to identify the top-k valuable users in a social network. The proposed methods take into account the interactions between users and network topology in weighted and directed graphs, and consider target users' profit and similarity. First, the influence graph is generated from the analysis of interactions and communications between users, which includes positive and negative influences among users. Then, the PMACO method is presented based on ACO by considering the profit of the selected nodes. To further improve the efficiency of the basic algorithm, the proposed IMOACO optimization model considers the similarity between nodes as well. This leads to a decrease in the redundancy between nodes, and the nodes with long distances are thus selected to cover the entire network, and the penetration rate of the spreading process increases. Extensive experiments are conducted on real and synthetic datasets. In later sections, we report a summary of the results, which show that the proposed algorithms based on ACO are capable of outperforming the state-of-the-art greedy algorithms.

The rest of this article is organized as follows. Section 2 reviews related studies, and the ant colony optimization is explained briefly in Section 3. Section 4 describes the proposed approaches applying the ACO algorithm to find valuable users. The results of evaluating the proposed approaches are reported in Section 5, and finally, the paper is concluded in Section 6.

2. Related works

Richardson and Domingos [26] pioneered research on selection of target nodes in social networks. Kempe et al. [16] tried to solve the influence maximization problem, and formulated it as a discrete optimization problem based on an independent cascade model. It has been shown that computation of the influence spread of a given node set is NP-hard [27]. Some of the works mentioned above have used Monte Carlo simulation, which gives an estimate of the influence that is spread by the seed set, but even a single run of the algorithm is very time-consuming.

Leskovec et al. [17] presented an optimization scheme that is known as Cost-Effective Lazy Forward selection (CELF) optimization, in which the expansion of each node is first calculated, and only the expansion of a limited number of nodes should be updated in subsequent iterations based on the expansion function. This makes CELF much faster than the greedy algorithm, because it significantly reduces the number of repetitions. Despite this optimization, the CELF algorithm still requires about 700 replications, which makes it slow and not applicable to large graphs. Goyal et al. [28] proposed a faster, more efficient version of the CELF algorithm, called CELF^{++}, which reduces the number of iterations of the algorithm. Chen et al. [29] concluded that computation of the spread based on the LT diffusion model in a general graph is an NP-hard problem, and it can be performed in linear time on directed acyclic graphs (DAGs). They assumed that each node can influence a limited number of its neighbors. Therefore, a local DAG (LDAG) is considered for each node, and the influence of that node is examined only for that local DAG. However, there are some limitations on their idea. For example, finding LDAGs constitutes an NP-hard problem. Therefore, a greedy exploration has been used to discover a good LDAG. Using a greedy LDAG leads to loss of quality in the seed set. In this method, it is assumed that a node can influence other nodes only through its paths in an LDAG, so its influence through other paths is ignored. It requires large amounts of memory in large networks to store all LDAGs. This algorithm is highly efficient in maximizing influence under the LT diffusion model, and is faster than CELF, but does not offer high-quality nodes. In Ref. [4], a community-based algorithm named CGA was proposed, which can be applied to both the LT and the IC models. A scalable algorithm referred to as State Machine Greedy Algorithm (SMG) was proposed by Heidari et al. in Ref. [30]. In this algorithm, the calculation made for counting the traversing nodes in the estimate propagation procedure is reduced through simplification of the Monte Carlo graph construction.

In the Simpath method [31], the vertex cover of the graph is computed to avoid calculation of the spread of each node in the graph, and the expansion of each node in the vertex cover is computed. Simpath investigates the probability of activating nodes by examining the paths that exist between the set of seed nodes and the other nodes on the input network. This algorithm searches all the paths from the initial seed to maximize influence in the LT-diffusion model. The problem of finding a simple path is NP-hard [32]. Simpath uses the vertex cover method to reduce the number of repetitions, and the resulting nodes are audited to maximize penetration. In this method, the number of selected nodes increases after each repetition of the algorithm. Therefore, the process of influence maximization is slow. In general, these methods use greedy algorithms that may not reach the global optimum. Tang et al. [33] proposed an influence maximization algorithm called TIM (Two-phase Influence Maximization) that incorporates novel heuristics, and enhances the greedy algorithm through constant-factor approximation, but is constrained by a specific seed set size. To address this issue, the Sketch-Based Influence Maximization and Computation (SKIM) algorithm was proposed by Cohen et al. in Ref. [34], which uses sketched influence paths for nodes. Thus, based on the sketch, the nodes with maximum influence are selected as the seed set members to accelerate the influence spread. In Ref. [34], another improvement for TIM was proposed exploiting estimation techniques based on martingales to reduce memory consumption. Although greedy algorithms provide good approximation guarantee, they suffer from high complexity [34,35] and low scalability [30].

Since metaheuristic approaches give better results than greedy, deterministic, and stochastic approaches, Yang et al. proposed an ACO

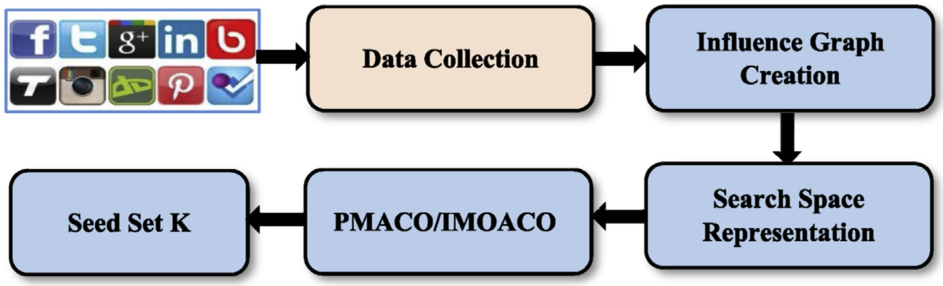


Fig. 1. Framework of the proposed method.

algorithm based on a parameterized probabilistic model in Ref. [24] to address the influence maximization problem. This work assumed that a company has a fixed budget for targeting *k* consumers who will trigger a cascade of influence. They used the degree centrality, distance centrality, and simulated influence methods for determining the heuristic values. In Ref. [15], a Cuckoo Search Diffusion Model (CSDM) was proposed that is based on a metaheuristic approach known as the Cuckoo Search Algorithm. Wang et al. [36] proposed a set-based coding genetic algorithm (SGA), which converges in probability to the optimal solution in target set selection problems. Furthermore, an approach based on Simulated Annealing (SA) was presented in Ref. [37] for the influence maximization problem, which applied two heuristic methods to accelerate the convergence process of SA, along with a new method of computing influence to speed up the proposed algorithm. The authors of [38] proposed a metaheuristic algorithm that can select the optimal seed set from the results of the NSGA-II algorithm. They considered both coverage size and diffusion time as the objectives of optimization in the diffusion process.

3. Ant colony optimization

The ant colony optimization (ACO) algorithm is one of the swarm intelligence methods, which is based on a probabilistic technique for solving computational problems. The ant colony algorithm that was initially proposed by Marco Dorigo in 1992 [39] constitutes some metaheuristic optimization. A near-optimal path in a graph is obtained based on the behavior of ants searching throughout their colony for food sources, and this is the main objective of the algorithm. In the natural world, ants search arbitrarily until they find food, and then return to their colony while pouring some pheromone on their trails. The next ants traverse a path with maximum pheromone. Due to the evaporation of the pheromone trail, more pheromone evaporates on long paths. Thus, pheromone density becomes higher on shorter paths than on others. If there were no pheromone evaporations, the ants would tend to be attracted to the path chosen by the first ant. Under such conditions, exploration of the solution space is constrained, which leads to convergence to a locally optimal solution. Given the above description, for solving an optimization problem using the idea of ant colony behavior, the search space is represented by a graph, and "simulated ants" traverse it to find a solution.

Many combinatorial optimization problems can be solved through application of the ant colony optimization method. ACO has simple implementation and lower execution time than most swarm intelligence methods, including the genetic algorithm and simulated annealing, in optimization problems such as feature selection and classification. Moreover, when the graph changes dynamically (e.g., in network routing and urban transportation systems), ACO can be applied continuously and adapted to the environment.

4. Proposed method

A social networking service/website such as Facebook or Instagram is an online platform that people use to build https://en.wikipedia.org/wiki/Social_networksocial relations with other people who have similar personal or professional interests, activities, backgrounds, or real-life connections. In this section, the steps of the proposed method are described (Fig. 1). First, the required data are collected from social networking websites, and the influence graph is generated given the negative effects from the communications and interactions among users in the collected data obtained in the previous step, and weak users are then excluded from the influence graph. In Fig. 2(a), an example of a synthetic influence graph with six nodes is shown, which is a weighted directed graph. Afterwards, the most important users in the influence graph are identified in the search space through employment of a modified ant colony optimization algorithm (PMACO/IMOACO). In this step, a given number of ants scroll the search space, and each ant selects the next node with the goal of *profit* maximization by applying the *state transition rule* and the *pheromone update rule*. The process of selecting nodes continues until the desirable set of nodes is selected. In the proposed method, the profit is the difference between the cost of the initial

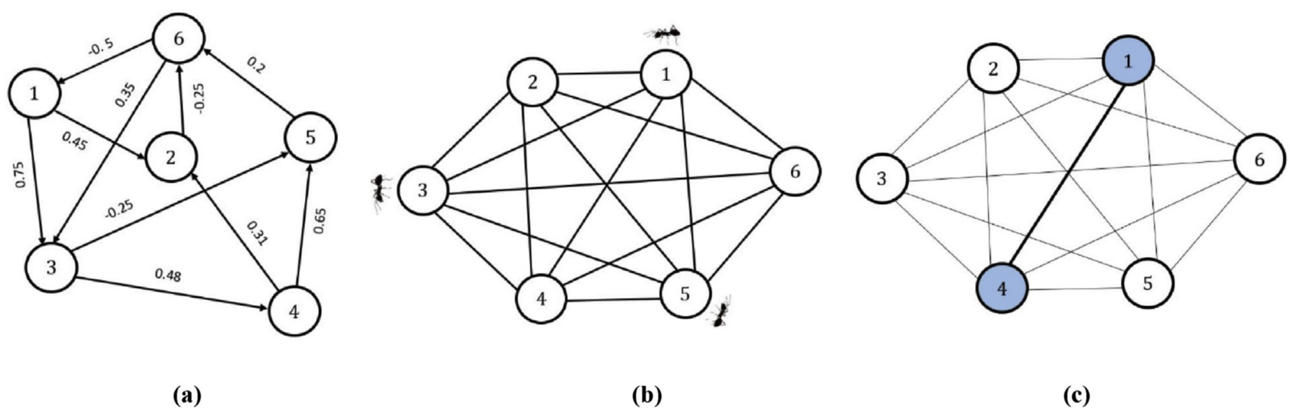


Fig. 2. (a) Influence graph (b) Ants move on the search space (c) Select seed set ($\theta = 0.5$).

activation (selected nodes) and the number of nodes activated by the selected nodes under a threshold value θ in the influence graph. For example, assume that we want to select two nodes from the influence graph as indicated in Fig. 2(a), where $\theta = 0.5$, and the cost of initial activation is zero.

In Fig. 2(b), an ant starts moving from node 1 in the first iteration; therefore, it selects this node (seed set = {1}), and then tries to choose the next node, which can be any other node. Assume that the ant selects node 5; according to the influence graph in Fig. 2(a), the out-neighbors of the new seed set {1, 5} that can be activated are nodes 2, 3, and 6. Since the weight of the edge between node 1 and node 2 ($W_{12} = 0.45$) is less than θ, node 2 is not activated. Similarly, node 6 is not activated either ($W_{56} = 0.2 < \theta = 0.5$). However, node 3 is activated because $W_{13} = 0.75 > \theta = 0.5$. If node 5 is selected, therefore, the profit will be equal to 1, because only one node will be activated. Alternatively, assume that the ant mentioned above selects node 4 (instead of node 5), and the seed set is changed to {1, 4}. Again, given the influence graph, the nodes that can be activated are 2, 3, and 5 in the next step, because $W_{13} = 0.75 > \theta$, $W_{45} = 0.65 > \theta$, and $W_{12} + W_{42} = 0.76 > \theta$, and the profit is equal to 3.

to represent the strength of influence.

The power of influence is obtained through analysis of relationships and interactive information. In some social networking websites such as Amazon and Epinion, users interact with each other in a variety of ways. For example, they create relationships by observing the records of other people's products or participating in the online discussion forums. In these networks, each user has a list of trusted friends, which includes a set of users whose records, articles, and statements are valuable to him/her. Similarly, a user may have a list of untrustworthy (blocked) users, whose comments are assumed worthless, offensive, or/and mistaken.

Another category of interactions involves rating and commenting on others' reviews, so that when a user posts a review on a product, other users may comment on or vote for it by simply clicking on a "yes/up" or "no/down" button. Techniques for opinion mining and emotion analysis can be applied to automatically determine the tendency of the comments and derive the influence relationship between users. With the above issues integrated and merged, the influence relations are regulated in Eq. (1).

$$W_{uv} = \begin{cases} +1 & \text{if user } u \text{ is in the trust list of user } v \\ -1 & \text{if user } u \text{ is in the block list of user } v \\ (1 - e^{-x})/(1 + e^{-x}) & \text{if user } v \text{ comments on or rates useru's review} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The ant tries other nodes, and pours some pheromone on the edges according to the profit values. This process is performed for other nodes, and the profit value of selecting node 4 is maximal, and the final seed set is therefore {1, 4} according to Fig. 2(c). This process is repeated by other ants in several iterations. Details on each step in Fig. 1 are described below.

4.1. Data collection

Information about user profiles and the effective communications between them on social networks can be collected from social network websites through crawlers. This information includes user profile information such as user accounts, occupations, and genders. Social relations among users such as Facebook friendships, Twitter followers, and trust/distrust in Epinion represent some sort of influence among users. Moreover, on some social networking websites, users can rate each other's comments. For example, users can rate comments at different levels: 1) not useful 2), fairly useful, 3) useful, 4) very useful, and 5) excellent. This rating is intended to evaluate users' views, and also reflects the relationships of influence among users. For example, where user *A* always has a high rating for user *B*, user *A* trusts user *B*, and user *B* is likely to have a great impact on user *A*. Furthermore, the interactions between users contain supportive and dissuasive information. Extraction of this information using emotion analysis techniques can reveal the relationships of influence among users.

4.2. Generation of the influence graph

The influence graph is generated through analysis of the collected data. In social networks, there exist relations of influence among users, and the weights of the connections between different pairs of users may be different. Since influence relationships are asymmetric, the network of users is displayed as a directed graph to better illustrate the nature of the influence relationships. In the influence graph *G (V, E)*, *V* is the set of all users (with the total number of *n*), and *E* is the set of influence relations. A directed link from node *u* to node *v* is represented as (*u, v*), where node *u* is the parent node, and *v* is the child of *u*. Each directed link has a weight

where *x* is the difference between the total number of positive comments or ratings from user *v* on user *u* and the total number of negative comments or ratings from user *v* on user *u*. Finally, if the sum of weights of the out-neighbors of a node is less than a specified threshold value, the node will be known as a *weak-node*, and is discarded from the influence graph.

4.3. Similarity between users

In the process of viral marketing aided by social networks, the similarity between users is of great importance in terms of the coherence of the users who are activated by them given users' interactions and communications. Therefore, the target users should be selected in such a way that they have the least possible similarity, in the sense that they are as far apart as possible, given maximum coverage of the network. Another issue concerns the speed of propagation over the network and increase in the speed of the diffusion process. Thus, if two users are similar in terms of power expansion at any time instant, it will be better to select only one of them. In the example shown in Fig. 3, two users A and B can influence four and five users, respectively, and user C affects three users. The expansion rate for both users A and B is greater than that of user C. If these two users are selected as influential nodes, however, they can maximally activate five users. Nevertheless, if we choose users A and C as target users, seven users will be influenced generally. As a result, selection of users with less similarity between them can lead to an increase in network expansion.

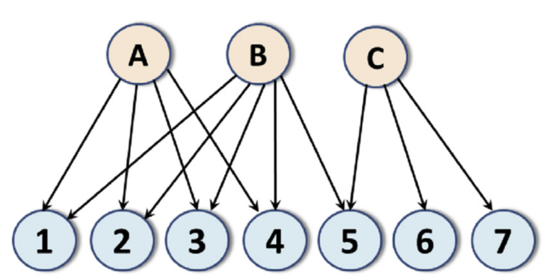


Fig. 3. Similarity effect on influence power between users.

It is also important to consider the similarity between users in terms of delay in activation. That is, selection of users who are topologically far from each other (with less similarity) can increase propagation speed and reduce activation delay significantly. In our multi-objective model, therefore, we set the level of similarity among users as the second objective, in the sense that users are selected as target who are least similar to each other.

In order to calculate the similarity of the nodes in the weighted directed graph obtained in the previous steps, we propose a semi-local weighted method. Given that the influence of each node decreases gradually as levels get higher, only two levels of neighboring users are considered for calculation of the similarity of nodes. The similarity value of node u to node v is measured according to Eq. (2).

$$Similarity(u,v) = \begin{cases} 1 & u \in \Gamma_1^{out}(v) \\ \alpha \dfrac{\left|\Gamma_1^{out}(u) \cap \Gamma_1^{out}(v)\right|}{\left|\Gamma_1^{out}(u) \cup \Gamma_1^{out}(v)\right|} + \beta \dfrac{\left|\Gamma_2^{out}(u) \cap \Gamma_2^{out}(v)\right|}{\left|\Gamma_2^{out}(u) \cup \Gamma_2^{out}(v)\right|} & \end{cases} \tag{2}$$

where $\Gamma_1^{out}(u)$ is the set of first-order neighbors of u (direct neighbors), and $\Gamma_2^{out}(u)$ is the set of second-order neighbors of u (neighbors of neighbors). $\left|\Gamma_1^{out}(u) \cap \Gamma_1^{out}(v)\right|$ is the number of common first-order neighbors of nodes u and v.

Since the graph is directed, the similarity between nodes is not symmetric. In other words, *Similarity* $(u, v) \neq$ *Similarity* (v, u). If node u is a member of $\Gamma_1^{out}(v)$, therefore, its similarity value to v is Similarity $(u, v) = 1$, because node u naturally affects all nodes influenced by v given that the graph is directed, and node u has influence on node v. However, Similarity (v, u) may not be equal to 1, and can be any value between 0 and 1. Generally, as given in Eq. (2), we consider the similarity of two nodes as the weighted sum of the number of common first-order neighbors in proportion to the total number of neighbors at the first level and the number of common second-order neighbors in proportion to the total number of neighbors at the second level. α and β $(\alpha + \beta = 1)$ have values between 0 and 1, and are used for weighting the first and second levels, respectively. Since the direct impact of users on each other is higher, we set the values to $\alpha = 0.7$ and $\beta = 0.3$. Finally, the similarity between the selected nodes is defined in Eq. (3).

$$S = \sum_u \sum_v Similarity(u,v) k_u k_v \tag{3}$$

where $k_u = 1$ when node u is selected as a member of the seed set K, and $k_u = 0$ otherwise. Consequently, S is the similarity value between the members of the subset of nodes selected as the target set, which should be as low as possible.

4.4. Representation of the search space

For application of the ACO algorithm, the search space should be appropriately represented for the node selection process. This is one of the basic steps in the proposed method. Therefore, we offer a fully-connected directed weighted graph $G = (N, E, W)$ for search space representation, where $N = \{N_1, N_2, \ldots, N_n\}$ is the set of all the n nodes, in which each user is represented by a node, $E = \{(N_i, N_j) : N_i, N_j \in n\}$ is the set of graph edges, and W is the set of weights of the edges, which contains the similarity values between nodes, obtained from Eq. (2).

Fig. 4 depicts the representation of the user selection problem, in which the set of nodes is $N = \{N_1, \ldots, N_6\}$, and $S_{1.6}$ shows the similarity of node N_1 to N_6. One of the advantages of such representation is that there is no limitation for ants in selection of nodes, which causes them to have the highest degrees of flexibility. The ants can move from anywhere in the graph and select different nodes in different positions.

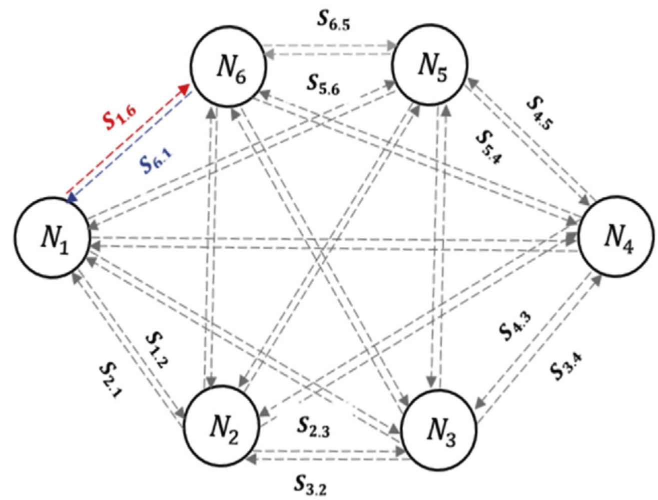


Fig. 4. Graph representation for the node selection problem.

4.5. Operation of the proposed method

Heuristic information should be defined as the main set of parameters in the ACO algorithm. In the proposed method, for PMACO, exploratory information is simply determined by the profit obtained from a set of nodes, and for IMOACO, heuristic information is a combination of profit and the inverse similarity between nodes, which corresponds to the weights of the edges in the graph. In the simulation of the proposed ACO-based algorithm, the influence graph and the similarity matrix of the nodes obtained from the preceding steps are given as input to the problem. The number of repetitions of the algorithm, the number of ants that are supposed to be placed on the search space, and other parameters are also given as inputs to the problem. In general, the proposed method consists of three phases, which are described below. The pseudo-code is shown in Algorithm 1.

- In the first phase (lines 2 and 3), the similarity between nodes is calculated and assigned to the corresponding edges in the represented graph. Moreover, the amount of initial pheromone on each edge is set to a constant value τ_0.
- The second phase of the proposed method (lines 4 to 12) addresses the process of allocating pheromone onto the edges in the ACO algorithm. This part consists of successive repetitions, at the beginning of which a random number in range $[1 \ldots n]$ is generated for each ant. That is, all ants start their trips on the search space randomly. This increases global search capability, and avoids falling into a local optimum. Each ant should traverse a path of length K-1, where K is the size of the set of influence nodes (seed set). Afterwards, each ant scrolls the graph nodes sequentially in order to select the next node by applying the state transition rule. This rule is defined such that the ants choose the path with the highest value of pheromone containing the nodes with the highest profit and the least similarity to each other. The process of node selection by each ant continues until the desirable K nodes are selected. After that, the path traversed by each ant is evaluated with the fitness function, which is described in Section 4.7. Finally, the pheromone values of the edges are updated through application of the pheromone updating rule at the end of each iteration, as detailed in Section 4.8. In other words, some of the pheromone corresponding to the edges evaporates, and an amount of pheromones is then added to the edges depending on the value of fitness. In fact, the ants try to place more pheromone on the edges with higher levels of fitness. This process is repeated for a given number of iterations.
- The third phase of the proposed method (lines 13 to 17) involves selection of the final subset of nodes. In each iteration described in the

previous step, the amounts of fitness of the paths traversed by the ants are compared, and the path with the highest fitness value in that iteration is stored. Eventually, the nodes on that path are described as making up the final subset of nodes.

Algorithm1. PMACO/IMOACO pseudo code

Input: G:Influence graph
S:Similarity matrix
I:Number of iterations
A:Number of ants
K:Size of seed set
τ_0:The initial amount of pheromone
Output: S:Seed set by size K
Begin algorithm
Apply the similarity $S_{i,j}$ between nodes, $\forall\, i,j = 1\ldots n$.
$\tau(i,j) = \tau_0, \quad \forall\, i,j = 1\ldots n$. /* initial pheromone */
  For $t = 1$ *to* I
    For $i = 1$ to A
      Place ant_i randomly on the graph nodes.
        For j = 1:K-1
          Choose the next node according to state transition rule Eq. (5).
        End For
      Compute profit and similarity for the ant_i tour according to Eq. (6) and Eq. (7) respectively.
    Compute fitness for the ant_i tour according to Eq. (8).
  End For
  Update pheromones according to Eq. (9).
  Sort ants according to their corresponding fitness.
End For
Return seed set K as final selected nodes.
End algorithm

4.6. State transition rule

To select the next suitable node, the ants follow the state transition rule. This rule is based on a combination of exploratory information and pheromone values, and is expressed as follows: When ant m is on node i, the next node j can be picked up through application of either the greedy or the probabilistic rule. In the greedy rule, the next node is obtained according to Eq. (4).

$$j = \underset{u \in J_i^m}{\arg\max}\left[(\tau_{iu})^{\eta}.\Pr\left(\Gamma_{t-1}^m, \mathrm{N_u}\right)^{\psi}.\left(1/S\left(\Gamma_{t-1}^m, \mathrm{N_u}\right)\right)^{\gamma}\right] \tag{4}$$

where J_i^m is the set of nodes that have not been selected yet. τ_{iu} indicates the value of pheromone on edge (i, u). Γ_{t-1}^m represents the set of nodes selected by ant m in the previous step $(t-1)$. $\Pr(\Gamma_{t-1}^m, N_u)$ is the profit obtained from a set including the nodes selected in step $t-1$ as well as node u, shown as N_u (Eq. (6)), and $s(\Gamma_{t-1}^m, N_u)$ is the similarity value between the set including the nodes selected in step $t-1$ to node u (Eq. (7)). Parameters η, ψ, and γ are constant values that are used to control the importance of pheromone trail versus heuristic information, including profit and similarity between nodes. In the probabilistic rule, on the other hand, ant m selects the next node j according to Eq. (5), defined as follows.

$$P_m(i,j) = \begin{cases} \dfrac{(\tau_{ij})^{\eta}.pr\left(\Gamma_{t-1}^m, \mathrm{N_j}\right)^{\psi}.\left(1/S\left(\Gamma_{t-1}^m, \mathrm{N_j}\right)\right)^{\gamma}}{\sum\limits_{u \in J_i^m}(\tau_{iu})^{\eta}.pr\left(\mathrm{N_i}, \mathrm{N_u}\right)^{\psi}.\left(1/S\left(\mathrm{N_i}, \mathrm{N_u}\right)\right)^{\gamma}}, & if\ j \in J_i^m \\ 0, & otherwise \end{cases} \tag{5}$$

It can be concluded that ants can pick up the best node greedily by applying the transition rule; otherwise, each node can have a chance to be selected based on the probability calculated for it.

4.7. Fitness

After the process of path selection, the suitability of each path scrolled by ant m is calculated according to a fitness function. In the proposed method, we consider a combination of profit and similarity of the nodes on each path as the *fitness* function, such that the profit value Pr_{Γ_m}, obtained from the nodes on the path, is computed using Eq. (6) as follows.

$$pr_{\Gamma_m} = r\sum_{u \in V} y_u - cK$$
$$y_u = \begin{cases} 1 & \sum\limits_{v \in \Gamma_m} W_{vu} \ge \theta \\ 0 & otherwise \end{cases} \tag{6}$$

where r is the revenue from buying a product. The value of y_u is 1, if node u buys products under the influence of the nodes that are selected by ant m; otherwise, it is zero. c is the cost of selecting a node as influence node, and K is the size of the seed set. W_{vu} is the weight of the edge between node v and node u, Γ_m is the set of nodes selected by ant m, and θ is the threshold value. In addition, the mutual similarity S_{Γ_m} between the selected nodes on the path traversed by ant m is calculated through Eq. (7).

$$S_{\Gamma_m} = \sum_u \sum_v similarity(u, v) K_u K_v, \qquad u, v \in \Gamma_m \tag{7}$$

where K_u is 1 if node u is a member of the set of selected nodes that exist on the path traversed by the ant, $(u \in \Gamma_m)$; otherwise, it is zero. Finally, we obtain the fitness function of the path from the mixture of these two functions, as shown in Eq. (8).

$$fitness_{\Gamma_m} = (pr_{\Gamma_m})^{\psi}.\left(\frac{1}{S_{\Gamma_m}}\right)^{\gamma} \tag{8}$$

4.8. Pheromone update

When all the ants complete their routes, the pheromone updating rule is used to update the pheromone values corresponding to the edges. This rule is applied to all edges when all the ants complete their paths. The pheromone updating rule is shown in Eq. (9).

$$\tau_t(i,j) = (1-\rho)\ \tau_{t-1}(i,j) + fitness_{\Gamma_m} \tag{9}$$

where $\tau_t(i,j)$ and $\tau_{t-1}(i,j)$ are the amounts of pheromone on edge (i, j) at times t and $t-1$, respectively. ρ is the pheromone evaporation coefficient.

5. Experiments and results

In this section, several experiments are carried out on real and synthetic datasets, and the results are reported and analyzed. For evaluation of the proposed method, it is compared with a number of well-known algorithms, including *High Weighted Degree* [40,41], *Weighted Closeness* [41], *CELF* [17], *LDAG* [29], *Simpath* [31], *GLR* [13], and *Degree Punishment (DP)* [14]. Details on the experiments, including information about the datasets, the generation of the influence graph from datasets, parameter setting, numerical results, sensitivity analysis of the parameters, and the complexity analysis are explained below.

5.1. Datasets

For assessment of the proposed method, three datasets are used in this paper. The Extended Epinions collection, compiled by Paolo Massa from the Epinions website, contains three files: 1- User rating, 2- Mc, and 3- Rating. The first file includes trust and distrust lists among users, the second file contains information on the authors of the articles, and the third file consists of qualitative sentences and ratings given by users ac-

Table 1
Scaling the rate values.

Scale setting	Rating
1	5
0.75	4
0.25	3
−1	2
−1	1

cording to the quality of the concept and content. The second real dataset is the Bank Wiring Room, which is a weighted directed network [42]. An artificial dataset with 64 nodes, which is generated using the LFR (Lancichinetti-Fortunato-Radicchi) [24] network model, is another dataset used in our experiments. The tuning parameters in the LFR model include average degree k, which is equal to 4, maximum degree, which is 8, mixing parameter μ with the value of 0.3, minimum number of communities, which is equal to 2, and maximum number of communities, which is equal to 4. The weights of the edges between nodes in LFR network are assumed to hold random values between −1 and +1.

5.1.1. Generation of the influence graph from Extended Epinions

For generation of the influence graph from the Extended Epinions dataset, the data aggregated in the three files mentioned in the previous section are used. The *User rating* file includes several thousands of records related to users who either trust or distrust each other. If user u is on the trust list of user v, for instance, user v trusts user u, and the weight of the edge $w_{uv} = 1$. If user u is on the distrust list of user v, v does not trust u, and the weight of the edge $w_{uv} = -1$. These trust and distrust relations are considered as a part of the influence graph.

The *Mc* file contains several thousands of records, including information on the subjects, purposes, and authors of the articles. Furthermore, the *Rating* file contains information on the purposes of the articles and the users who have rated the articles, with the rating value between 1 and 5. To find the influence value between users using these collocated data, we need to extract the hidden effects between the users. For this purpose, we use the information in the Mc and Rating files. As mentioned previously, the Mc file contains information on the subject and the starred author (corresponding author) of the article. In the Rating file, on the other hand, the subject of the article is given as well as the rating number. From the combination of this information, we form a directed graph that includes the voting user, the user who is voted for (author of the article), and the vote value (1–5).

If a user gives the rating value 5 to an author, the weight of the edge between the two users is 1, the weight is 0.75 if the rating value is 4, the weight is 0.25 if the rating value is 3, and for rating values equal to 2 or 1, we consider the weight of the directed edge between the users to be −1 (Table 1). In the next step, the obtained graph is compared with the graph generated in the previous section, which contains the trust and distrust relationships between the users, and the duplicate edges are eliminated. If two users trust or distrust each other, the weight of the edge between them is fixed and assumed to be 1 or -1, respectively, and the ratings given by them to each other do not have much effect on their relationship. Since a user may be the corresponding author of several articles, there may be several edges between two users in the graph obtained from the previous steps. In a directed graph, however, there must be at most one edge between any two nodes in each direction. Therefore, we consider all directed and weighted edges with both positive and negative weights between two users, and count the positive and negative weights. Consequently, the final weight of the edge from user u to user v is calculated according to Eq. (10).

$$w_{uv} = \frac{1 - e^{-x}}{1 + e^{-x}}$$

$$x = \sum_{u} A^{+}_{uv} - \sum_{u} A^{-}_{uv} \quad , u \in V \tag{10}$$

Table 2
Bank Wiring Room properties.

Description	Edges	Nodes	Type	Network
Antagonistic (negative) behavior	19	14	Undirected	RDNEG
The number of times workers traded job assignments	26	14	Directed	RDJOB
Helping others with work	24	14	Directed	RDHLP
Participation in arguments about open windows	19	14	Undirected	RDCON
Friendship	13	14	Undirected	RDPOS
Participation in horseplay	28	14	Undirected	RDGAM

Table 3
Parameter setting.

Value	Notation	Parameter
50	I	Maximum number of cycles
[1 … 50]	K	Number of selected nodes
K	A	Number of ants
1	τ_0	The initial amount of pheromone
0.05	ρ	Pheromone evaporation coefficient
1	η	Relative importance of the pheromone value
1	ψ	Relative importance of the profit value
1	γ	Relative importance of the similarity value
1	θ	Threshold
0.1	c	Cost
1	r	Return from one customer

In Eq. (10), if there is an edge with positive weight from user u to user v, then $A^{+}_{uv} = 1$; otherwise, $A^{+}_{uv} = 0$. Similarly, if there is an edge with negative weight from user u to user v, then $A^{-}_{uv} = 1$; otherwise, $A^{-}_{uv} = 0$. Therefore, the weights between nodes are $w_{uv} \in [-1, +1]$, which shows the influence between the users.

5.1.2. Generation of the influence graph from Bank Wiring Room

The Bank Wiring Room (BWR) dataset was collected from observation of the social relations between a group of 14 Western Electric (Hawthorne Plant) workers (9 wiremen + 3 soldermen + 2 inspectors) for six months. It consists of six networks, each of which has 14 nodes. The networks have 13 to 28 edges; four networks are undirected, and two are directed. Details about the social interactions and relationships including friendships, participation in discussions, negative relationships, etc. are presented in Table 2. We use the information from the six networks mentioned above to generate the influence graph.

The RDNEG network consists of negative and opposite relationships observed between users. It is undirected with 19 edges. We change it to a directed graph, and the number of edges thus changes to 38. Since the network contains negative relationships, the weight of the edges is −1. Similarly, the RDPOS graph is undirected, and contains friendship relationships with 13 edges. Because friendship is naturally a two-way relationship, we change it to a graph with 14 nodes with 26 directed links, and add it to the previous graph to construct a part of the influence graph. Due to the friendship relations between nodes, the weight of these links is assumed to be +1. The RDJOB network is a directed network containing 26 links between employees who trade their proprietary tasks. The weights of these links are set to 0.25, and the network is added to the influence graph. The RDHLP network is directed, and concerns users who help each other. Again, the weights are set to 0.25. Both RDGAM and RDCON networks are directed and related to participation in discussions, games, and fun. The weights are also assumed to hold the value of 0.25, and these graphs are also added to complete the influence graph. Finally, there might be more than two edges between two employees, and we therefore compute the sum of the weight values of the edges. The influence graph resulting from the above-mentioned interactions is used in the next section with redundant edges removed.

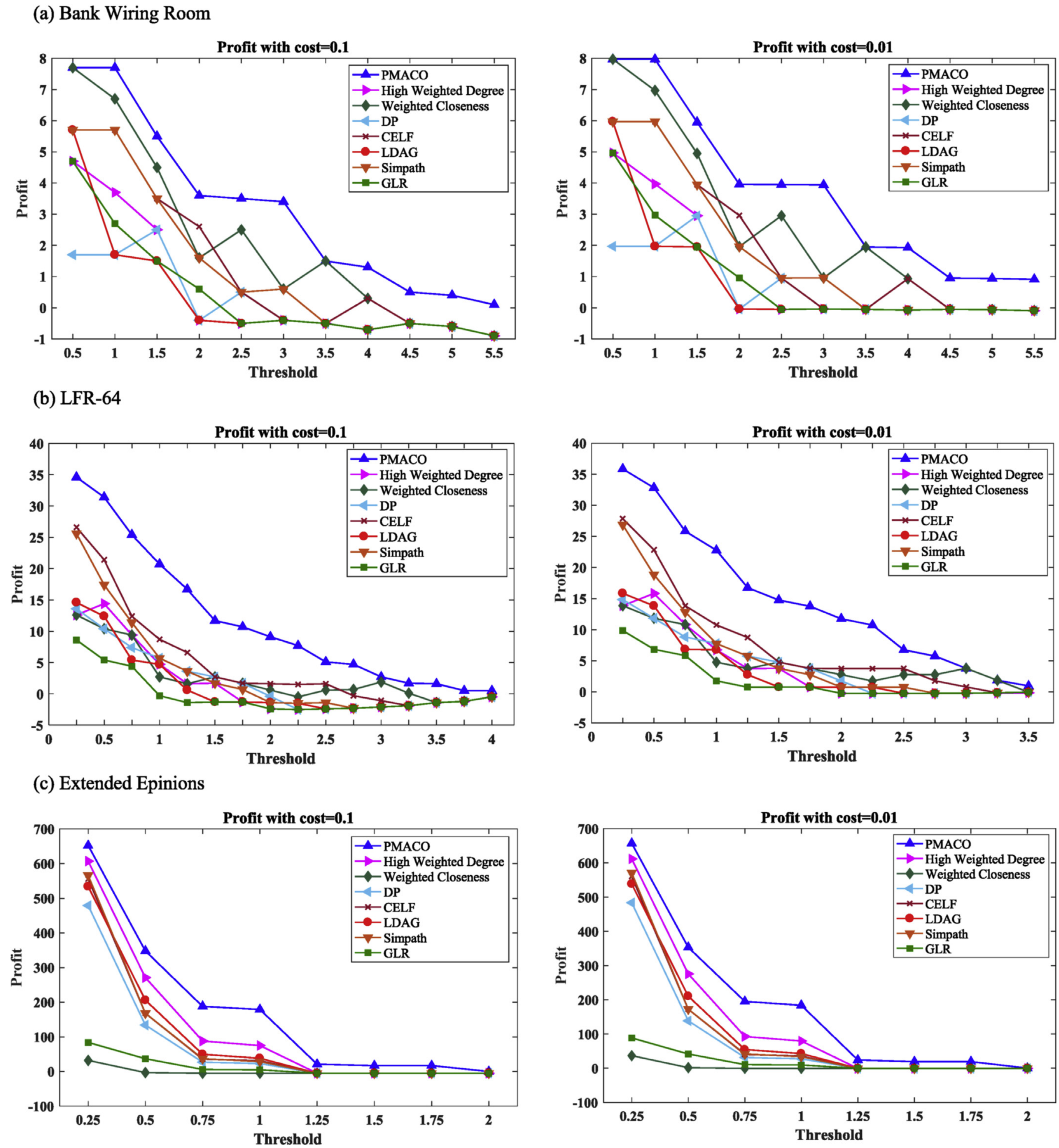


Fig. 5. The comparison of the profit obtained by the proposed method (PMACO) and the other algorithms.

5.2. Parameter setting

The proposed method contains various parameters, which must be tuned. Table 3 shows the values of the parameters. The number of ants (A) for each dataset is K (the number of selected nodes). Parameters such as η, ψ, and γ, which correspond to the weight of past experiences (pheromone) and exploratory information including the first objective function (profit) and the second objective function (similarity), are set. The value of threshold θ is set to 1, and is used in two parts of the proposed method: 1- computation of the profit obtained from nodes and 2- removal of weak-nodes from the influence graph, so that all nodes the sum of the weights of the outgoing edges of which is less than threshold θ are removed. During the program implementation of the proposed algorithms, the maximum number of iterations is set to $I = 50$, which is enough for the convergence. Pheromone evaporation rate is set to $\rho = 0.05$. The parameter τ_0, denoting the initial pheromone value, is set to 1. Moreover, LDAG Algorithm requires a parameter to control the size of LDAG which is set to 1/320 in this paper [29]. For Simpath and CELF,

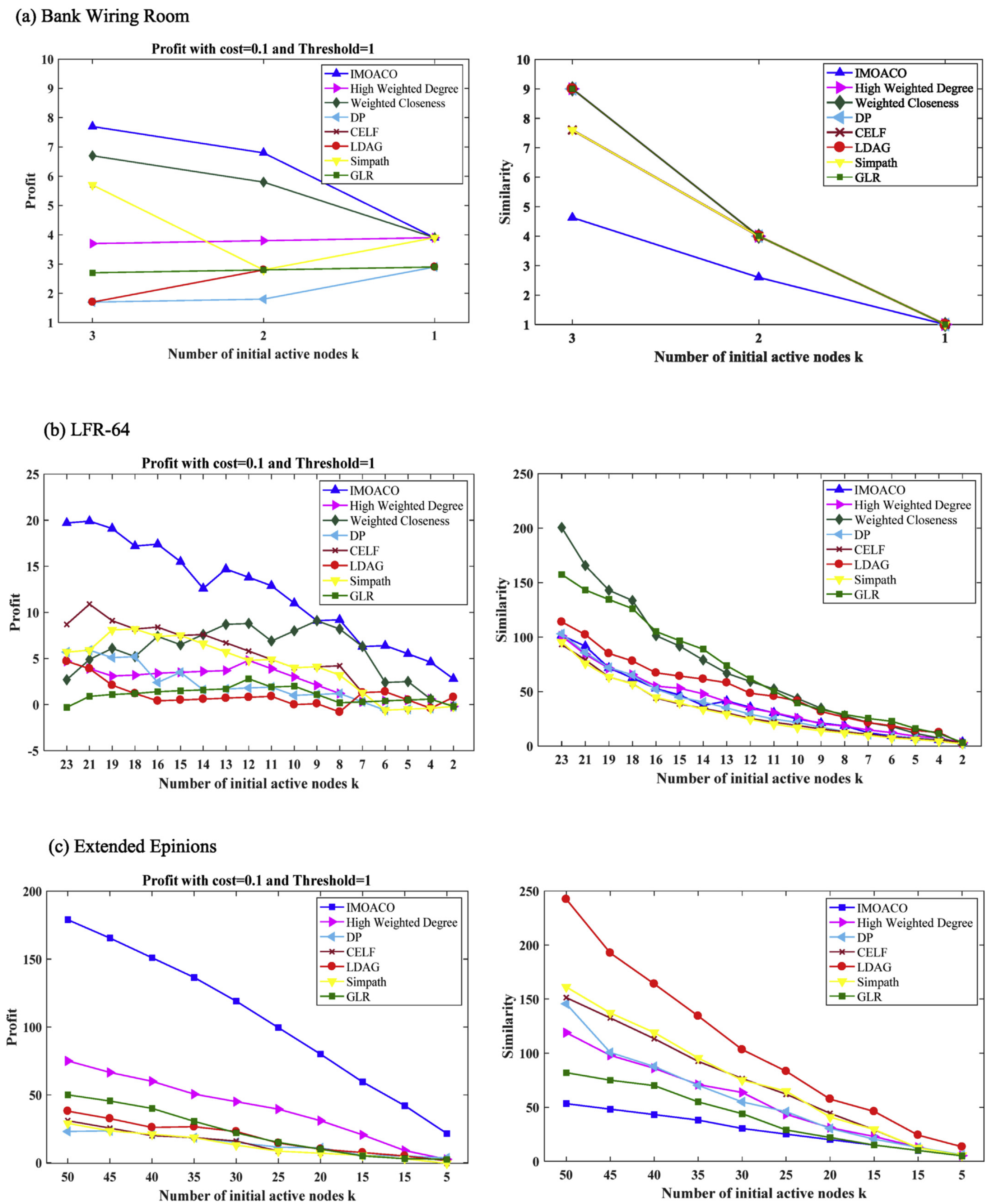


Fig. 6. The comparison of the profit and similarity values of the proposed method (IMOACO) and the other algorithms.

the pruning threshold for the backtrack method is set to 0.001, and the look-ahead value is set to 4 as the main paper [31]. In DP, the weaken factor ω is set to $\beta^c_{rand} \approx \frac{k}{k^2}$, which β^c_{rand} denotes the spreading threshold for the network [14]. The value of β^c_{rand} for Bank Wiring Room is 0.117, for the synthetic network generated by LFR with 64 nodes is 0.092, and for Extended Epinions is 0.096. The other parameters are set according to Table 3.

5.3. Numerical results

In this section, the proposed PMACO (Profit Maximization based on ACO) is first assessed on three datasets given only one objective aimed at maximizing profit. The datasets include Bank Wiring Room, LFR, and Extended Epinions. Then, the results of the proposed bi-objective algorithm IMOACO, aimed at maximizing profit and minimizing similarity in the selected seed set, are discussed and evaluated. In the case of the Extended Epinions dataset, we use the largest component, which has 15112 nodes and 53670 edges. Below, the proposed algorithm is compared with other algorithms, including High Weighted Degree, Weighted Closeness, CELF, LDAG, DP, GLR, and Simpath, in terms of target functions. Our experiments were executed on a PC with an Intel Core i7 CPU system with 2.7 GHz frequency and 8 GB RAM. We used Microsoft Visual Studio 2015 and the C# language to program the algorithms.

Fig. 5 shows the profits obtained for the proposed PMACO and other algorithms on different datasets. For the Bank Wiring Room graph, the size of the seed set K was varied in range [3,9]. As can be seen in Fig. 5(a), for all threshold values θ, the profit obtained by the proposed method is higher than those in the other algorithms when cost is 0.1. Furthermore, as the cost is reduced to 0.01, the proposed method yields more profit on this dataset. However, the profit increment is not significant. Moreover, Fig. 5(b) shows the profit values of the proposed method in comparison with those of the other algorithms on the LFR network with 64 nodes and different values of θ. As shown in the figure, the proposed method performs much better than the other algorithms for all threshold values. In other words, the profit gained from the selected nodes with the proposed method is much higher than the profit gained from the best nodes identified by the other algorithms. In Fig. 5(c), it is shown how the proposed method was applied to the largest connected component of the Epinions influence graph for the top 50 nodes (K = 50) detected by the proposed method and the other algorithms. The results demonstrate that the proposed method is superior to the algorithms mentioned above.

Fig. 6 depicts the results obtained by the proposed IMOACO in terms of target functions, separately and in comparison with the other algorithms on different datasets. As can be seen in Fig. 6(a), on the BWR dataset, the profits obtained from the selected seed sets of different sizes in the proposed algorithm are higher than those in the other algorithms (close to those in Weighted Closeness). Moreover, comparison of the similarity between the nodes selected with different algorithms demonstrates that the similarity between the selected nodes in the proposed method is less than that in the others. It is worth noting that the similarity values for the DP, LDAG, HWD, GLR, and WC algorithms are the same, and the points are overlapped. There are similar trends for the CELF and Simpath algorithms. This is because Bank Wiring Room in a small network with only 14 nodes and the likelihood of equality or proximity of the similarity values is very high for different algorithms.

Similarly, the values of the target functions of profit and similarity in the LFR network influence graph are compared in Fig. 6(b) with those in the other algorithms. As can be seen, the profit gained from the set of nodes identified by the proposed IMOACO method is more than that gained in the other algorithms. Moreover, as can be observed, the similarity of nodes in different sizes is less than that in the Weighted Closeness, High Weighted Degree, LDAG, and DP algorithms. In other words, the proposed method performs better than the algorithms mentioned above in terms of the degree of similarity between the selected nodes, while the other three algorithms, DP, Simpath and CELF, are close to the proposed approach (Simpath performs slightly better).

Fig. 6(c) shows the simulation results on the largest connected component of the influence graph from the Extended Epinions dataset for seed sets of different sizes from 5 to 50 identified by the proposed method and the other algorithms. As can be seen, there is a significant difference between the proposed method and the others in terms of the obtained profit. When the number of selected nodes is 50, there is a big gap between the proposed method and the other algorithms. Furthermore, the proposed method has much better performance than the other algorithms in terms of the degree of similarity between nodes.

As mentioned in Subsection 3.4, considering the similarity between nodes leads to increase in the influence spread of the selected nodes. In this experiment, the effect of similarity among users is evaluated based on the Independent Cascade (IC) model [43] on the Extended Epinions network. In the IC model, an inactive node v can be successfully activated by node u with a diffusion probability $p_{u,v}$ at time t. At each time step t, if node u has been newly activated at time step t − 1, it can make an attempt to activate every inactive neighboring node v with probability $p_{u,v}$. In other words, any node in the network tries to activate its inactive neighbors only at the time when it has just gotten activated. If the attempt is successful, node v will become active; otherwise, it will still stay inactive. In each case, node u can never try to activate other nodes at later time steps. If several newly activated nodes try to activate a specific node, the activation attempts are independent, and can proceed sequentially in an arbitrary order. If there are no more newly activated nodes in the network, the spreading process will terminate. Fig. 7 shows the results concerning the spreading capability of the selected seed sets of different sizes with the PMACO method, in which only profit is considered, in comparison with that in the IMOACO method, in which similarity is also taken into account. In this experiment, $p = 0.1$, and the edges with negative weights are ignored. As can be found from the figure, when similarity is considered in the selection process, the spread of influence is more than in the case where only profit is considered. The results were obtained by averaging over 100 simulation runs.

5.4. Sensitivity analysis of the parameters

There are some parameters that have an important impact on the performance of the proposed PIMACO algorithm, and control exploration and exploitation. The basic parameters that are used in this algorithm are the relative importance of pheromone η, the relative importance of the heuristic value ψ, and pheromone evaporation rate ρ. In this experiment, PMACO was tested on the Extended Epinions graph where $\theta = 0.25$, and the important parameters were tuned for their effect to be observed and their optimal values to be found. Accurate setting of these parameters substantially influences the results of the PMACO method, as will be

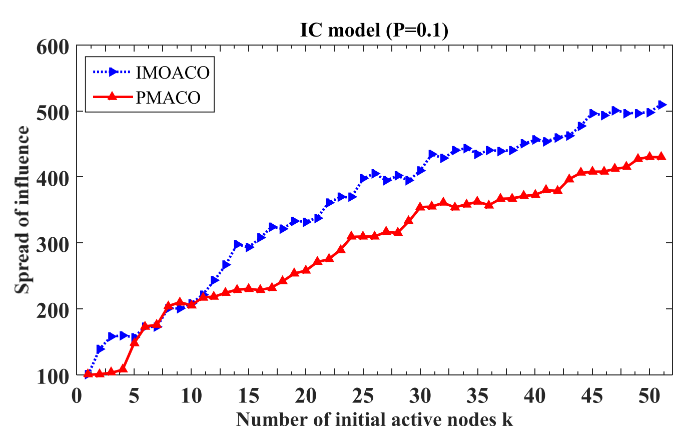
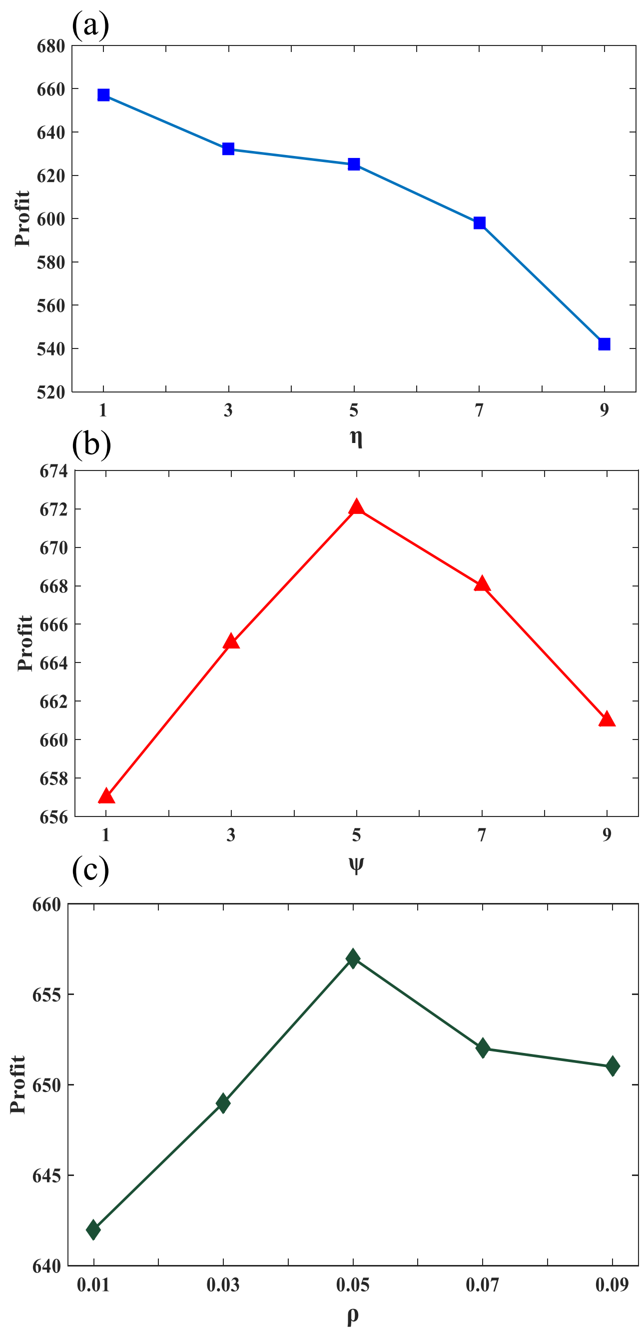


Fig. 7. The influence spreading of the proposed methods on Epinions under IC model with $p = 0.1$.

described below in detail. The values of the other parameters were set according to Table 3. As mentioned previously, parameter η demonstrates the importance of the pheromone collected on the edges. It is clear from Fig. 8(a) that most ants are inclined to select the path with more pheromone as η increases, so they much more likely follow the previously selected tour, which leads to lower performance required for obtaining the solutions. In this case, the best solution is for $\eta = \psi = 1$ and $\rho = 0.05$. In Fig. 8(b), the parameter ψ is justified. As the value of ψ increases, which indicates the importance of heuristic information, the ants pay less attention to the tours that have been traveled by other ants, which causes them to be trapped into local optima. Since the pheromone left by an ant on an edge evaporates with time, the parameter ρ affects exploration and randomness. Therefore, in the case where a large number of nodes exist, the pheromone on the edges becomes very low as evaporation rate increases, and they lose the chance of being visited by ants, and some nodes are therefore never traversed. According to the results shown in Fig. 8(c), the best solution for the proposed method is obtained where $p = 0.05$. In general, to obtain an optimal solution and avoid premature convergence, the algorithm must establish a balance between exploration and exploitation.

5.5. Time complexity analysis

According to the pseudo-code of the proposed method for PMACO, where only profit is considered as the main objective, time complexity can be estimated as follows. Suppose that N is the number of nodes in the influence graph. Since some of the nodes known as weak-nodes are removed in the last step of generating the influence graph, N is usually much smaller than the number of nodes n in the original social network ($N \ll n$). As can be seen in Algorithm 1, in the PMACO method, A ants



Network	#nodes	η	ψ	ρ	Profit
Epinions	15112	1	1	0.05	**657**
		3	1	0.05	632
		5	1	0.05	625
		7	1	0.05	598
		9	1	0.05	542

Network	#nodes	η	ψ	ρ	Profit
Epinions	15112	1	1	0.05	657
		1	3	0.05	665
		1	5	0.05	**672**
		1	7	0.05	668
		1	9	0.05	661

Network	#nodes	η	ψ	ρ	Profit
Epinions	15112	1	1	0.01	642
		1	1	0.03	649
		1	1	0.05	**657**
		1	1	0.07	652
		1	1	0.09	651

Fig. 8. The comparison of the profit obtained by PMACO with respect to different values for η, ψ and.ρ.

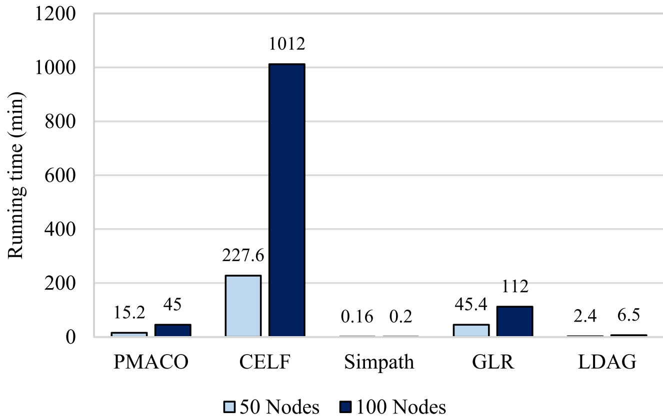


Fig. 9. Comparison of running time of different algorithms on Extended Epinions.

start to scroll the search space randomly from different points. Since the search space is a complete graph, each ant investigates all of the other nodes. This process is repeated as many times as the number of iterations I. Therefore, the time complexity of this part is O(IA (K-1) (N-1)), where K is the size of the seed set. It is worth mentioning that complexity can be reduced to O(I(K-1) (N-1)) if the ants run in parallel (multithreaded programming). In the next step, all the nodes are sorted based on their pheromone values with a cost of O(NlogN). Consequently, the complexity of the proposed PMACO is O(IA (K-1) (N-1) + NlogN = NlogN.

For the IMOACO method, the similarity between each two nodes is first evaluated with a computational cost of $O(N^2)$, and the other steps are similar to those in PMACO. Therefore, the complexity of this part is $O(N^2)$, where N is much smaller than n, as mentioned previously. In comparison with other methods, GLR detects two types of nodes including local critical nodes and gateway nodes based on the community structure of the network, and selects a subset of nodes as influential nodes which are close to the both mentioned groups. However, the community detection process in this algorithm is relatively time consuming. CELF reduces the number of calls to the spread estimation function, because only the spread of some nodes needs to be updated in each iteration, and the running time of the greedy algorithm is reduced. However, since the spread of every node is calculated in the first iteration, the time complexity of CELF is still high. In Simpath, for reduction of the number of calls to the influence estimation function, the vertex cover is first computed, and only the spread of nodes within the vertex cover is calculated. Given a set S of initial adopters, the influence of S, denoted as δ(S), is the expected number of adopters at the end of the diffusion process, subject to a specified diffusion model. The Simpath and CELF algorithms calculate $\delta(S_i + x + u) = \delta^{V-S_i-u}(x) + \delta^{V-x}(S_i + u)$ in each step. Here, S_i is the seed set after step i, and $i = |S_i|$. In order to identify seed $i + 1$, Simpath calculates both $(S_i + x)$ and δ^{V-x} $(S_i + u)$ simultaneously, and if x is chosen as a seed, only $\delta^{V-S_i-u}(x)$ will need to be calculated. In CELF, however, x is considered unknown in each iteration of the algorithm, and all the calculations are carried out multiple times [44]. Furthermore, Simpath uses a threshold value η to make a trade-off between influence spread and execution time, where a smaller η means greater influence spread and longer execution time. One of the advantages of the proposed method is that the profitable nodes are selected only from among the members of the influence graph. In other methods such as CELF, however, all of the nodes are considered in the calculations for identifying influential nodes in general social networks. In LDAG, a DAG is constructed for each node v, and the spread of nodes is computed locally within the resulting DAGs, and the seed nodes are selected based on the greedy algorithm. This algorithm has a reasonable running time, but it takes a lot of memory to store a DAG for each node in the graph. Fig. 9 shows the average runtimes (in minutes) of ten independent runs of the proposed PMACO compared to the CELF, Simpath, GLR, and LDAG algorithms on the Epinions network. Based on these results, the average execution time of the proposed method is much lower than that of CELF and GLR, but higher than those of Simpath and LDAG. According to the results, therefore, the main advantage of our approach is achievement of greater profit with low computational complexity.

6. Conclusion

Social influence has many applications such as marketing, advertisement, and recommendation. In this paper, single-objective and a multi-objective approaches to finding the most valuable users with maximum profit and minimum similarity between users were presented. In our ACO-based meta-heuristic methods, the influence graph that is derived from the interactions among users and also includes negative weights is traversed by ants. Each ant selects the next node such that maximum cumulative profit and the lowest similarity to the previously selected nodes is achieved. Therefore, if an edge (or ultimately a path) is chosen by many ants, the profit gained from selection of those nodes is high, and they are less similar to each other. Thus, such paths receive more pheromone, and their chance of being selected by other ants in the following iterations is higher. Finally, one would expect the best nodes to be selected in the proposed method. For validation, our method was compared with well-known algorithms for identification of influential nodes. The numerical results confirm the effectiveness of the proposed methods.

References

[1] C.-Y. Chiang, L.-H. Huang, B.-J. Li, J. Wu, H.-G. Yeh, Some results on the target set selection problem, J. Comb. Optim. 25 (2013) 702–715.
[2] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.
[3] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: European Conference on Principles of Data Mining and Knowledge Discovery, 2006, pp. 259–271.
[4] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 1039–1048.
[5] R. Narayanam, Y. Narahari, A shapley value-based approach to discover influential nodes in social networks, IEEE Trans. Autom. Sci. Eng. 8 (2011) 130–147.
[6] J. Tang, X. Tang, J. Yuan, Profit maximization for viral marketing in online social networks: algorithms and analysis, IEEE Trans. Knowl. Data Eng. 30 (2017) 1095–1108.
[7] K. Xu, J. Li, Y. Song, Identifying valuable customers on social networking sites for profit maximization, Expert Syst. Appl. 39 (2012) 13009–13018.
[8] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, T. Zhou, Identifying influential nodes in complex networks, Phys. A Stat. Mech. Appl. 391 (2012) 1777–1787.
[9] S. Gao, J. Ma, Z. Chen, G. Wang, C. Xing, Ranking the spreading ability of nodes in complex networks based on local structure, Phys. A Stat. Mech. Appl. 403 (2014) 130–147.
[10] W. Xu, W. Liang, X. Lin, J.X. Yu, Finding top-k influential users in social networks under the structural diversity model, Inf. Sci. 355 (2016) 110–126.
[11] Y. Zeng, X. Chen, G. Cong, S. Qin, J. Tang, Y. Xiang, Maximizing influence under influence loss constraint in social networks, Expert Syst. Appl. 55 (2016) 255–267.
[12] C. Salavati, A. Abdollahpouri, Z. Manbari, BridgeRank: a novel fast centrality measure based on local structure of the network, Phys. A Stat. Mech. Appl. 496 (2018) 635–653.
[13] C. Salavati, A. Abdollahpouri, Z. Manbari, Ranking nodes in complex networks based on local structure and improving closeness centrality, Neurocomputing 336 (2019) 36–45.
[14] X. Wang, Y. Su, C. Zhao, D. Yi, Effective identification of multiple influential spreaders by DegreePunishment, Phys. A Stat. Mech. Appl. 461 (2016) 238–247.
[15] N. Sinha, B. Annappa, Cuckoo search for influence maximization in social networks, in: Proceedings of 3rd International Conference on Advanced Computing, 2016, pp. 51–61. Netw. Inf.
[16] D. Kempe, J. Kleinberg, É. Tardos, Influential nodes in a diffusion model for social networks, Int. Colloq on Automata, Lang., and Program. (2005) 1127–1138.
[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 420–429.
[18] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 199–208.

[19] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.
[20] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, Int. Workshop Web Internet Econ. (2007) 306–311.
[21] E. Mossel, S. Roch, Submodularity of influence in social networks: from local to global, SIAM J. Comput. 39 (2010) 2176–2188.
[22] T. Cao, X. Wu, S. Wang, X. Hu, Maximizing influence spread in modular social networks by optimal resource allocation, Expert Syst. Appl. 38 (2011) 13128–13135.
[23] A. Borodin, Y. Filmus, J. Oren, Threshold models for competitive influence in social networks, Int. Workshop Internet Netw. Econ. (2010) 539–550.
[24] W.-S. Yang, S.-X. Weng, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Application of the ant colony optimization algorithm to the influence-maximization problem, Int. J. Swarm Intell. Evol. Comput. 1 (2012) 1–8.
[25] O. Cordón García, F. Herrera Triguero, T. Stützle, A review on the ant colony optimization metaheuristic: basis, models and new trends, Mathw. Soft Comput. 9 (2002).
[26] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 61–70.
[27] J.-R. Lee, C.-W. Chung, A query approach for influence maximization on specific users in social networks, IEEE Trans. Knowl. Data Eng. 27 (2014) 340–353, 1-1.
[28] A. Goyal, W. Lu, L.V. Lakshmanan, Celf++: optimizing the greedy algorithm for influence maximization in social networks, in: Proceedings of the 20th International Conference Companion on World Wide Web, 2011, pp. 47–48.
[29] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, Data Mining (ICDM), 2010, in: IEEE 10th International Conference on, 2010, pp. 88–97.
[30] M. Heidari, M. Asadpour, H. Faili, SMG: fast scalable greedy algorithm for influence maximization in social networks, Phys. A Stat. Mech. Appl. 420 (2015) 124–133.
[31] A. Goyal, W. Lu, L.V. Lakshmanan, Simpath: an efficient algorithm for influence maximization under the linear threshold model, Data Mining (ICDM), in: IEEE 11th International Conference on, 2011, pp. 211–220.
[32] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (1979) 410–421.
[33] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, 2014, pp. 75–86.
[34] E. Cohen, D. Delling, T. Pajor, R.F. Werneck, Sketch-based influence maximization and computation: scaling up with guarantees, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 629–638.
[35] S. Cheng, H. Shen, J. Huang, W. Chen, X. Cheng, IMRank: influence maximization via finding self-consistent ranking, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 475–484.
[36] C. Wang, L. Deng, G. Zhou, M. Jiang, A global optimization algorithm for target set selection problems, Inf. Sci. 267 (2014) 101–118.
[37] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, AAAI (2011) 127–132.
[38] A. Mohammadi, M. Saraee, Finding influential users for different time bounds in social networks using multi-objective optimization, Swarm Evol. Comput. 40 (2018) 158–165.
[39] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.
[40] T. Opsahl, F. Agneessens, J. Skvoretz, Node centrality in weighted networks: generalizing degree and shortest paths, Soc. Netw. 32 (2010) 245–251.
[41] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016) 1–63.
[42] R.L. Breiger, S.A. Boorman, P. Arabie, An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling, J. Math. Psychol. 12 (1975) 328–383.
[43] J. Goldenberg, B. Libai, E. Muller, Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata, Acad. Market. Sci. Rev. 2001 (2001) 1.
[44] K. Rahimkhani, A. Aleahmad, M. Rahgozar, A. Moeini, A fast algorithm for finding most influential people based on the linear threshold model, Expert Syst. Appl. 42 (2015) 1353–1361.