BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 04/05 pp100–117
Volume 2, Number 2, June 2019
DOI: 10.26599/BDMA.2018.9020034

Spreading Social Influence with both Positive and Negative Opinions in Online Networks

Jing (Selena) He*, Meng Han, Shouling Ji, Tianyu Du, and Zhao Li

Abstract: Social networks are important media for spreading information, ideas, and influence among individuals. Most existing research focuses on understanding the characteristics of social networks, investigating how information is spread through the "word-of-mouth" effect of social networks, or exploring social influences among individuals and groups. However, most studies ignore negative influences among individuals and groups. Motivated by the goal of alleviating social problems, such as drinking, smoking, and gambling, and influence-spreading problems, such as promoting new products, we consider positive and negative influences, and propose a new optimization problem called the Minimum-sized Positive Influential Node Set (MPINS) selection problem to identify the minimum set of influential nodes such that every node in the network can be positively influenced by these selected nodes with no less than a threshold of θ. Our contributions are threefold. First, we prove that, under the independent cascade model considering positive and negative influences, MPINS is APX-hard. Subsequently, we present a greedy approximation algorithm to address the MPINS selection problem. Finally, to validate the proposed greedy algorithm, we conduct extensive simulations and experiments on random graphs and seven different real-world data sets that represent small-, medium-, and large-scale networks.

Key words: influence spread; social networks; positive influential node set; greedy algorithm; positive and negative influences

1 Introduction

A social network (e.g., Facebook, Google+, and MySpace) is composed of a set of nodes (such as individuals or organizations) that share a similar interest or purpose. The social network is a powerful medium of communication for sharing, exchanging, and disseminating information, and for spreading influence beyond the traditional social interactions. Since social networks emerged, they have significantly enpanded our social circles and become a bridge to connect our daily physical life and the virtual web space. With the emergence of social applications (such as Flickr, Wikis, Netflix, and Twitter, etc.), a tremendous interest has focused on how social networks can be utilized effectively to spread ideas or information within a community[1–6]. Capturing the dynamics of a social network is a complex problem, thus, it requires an approach to analyze the dynamics of positive and negative social influences that result from individual-to-individual and individual-to-group interactions. Individuals in a social network may have both positive and negative influences on each other. For example, within the context of

• Jing (Selena) He and Meng Han are with the College of Computing and Software Engineering at Kennesaw State University, Kennesaw, GA 30144, USA. E-mail: {she4, mhan9}@kennesaw.edu.
• Shouling Ji and Tianyu Du are with the Department of Computer Science at Zhejiang University, Hangzhou 310058, China. E-mail: {fsji,zjradtyg}@zju.edu.cn.
• Zhao Li is a senior staff scientist and director at Alibaba Group, Hangzhou 310052, China. E-mail: lizhao.lz@alibaba-inc.com.
* To whom correspondence should be addressed.
Manuscript received: 2018-07-20; accepted: 2018-09-19

gambling, a gambling insulator has a positive influence on his friends/neighbors. Moreover, if many of an individual's friends are gambling insulators, then the aggregated positive influence is exacerbated. However, an individual might become a gambler, who has a negative impact on his friends/neighbors. For example, in the social network shown in Fig. 1, social influences are represented by weights assigned to edges. If Jack and Bob (marked by the person with a red tie) are gambling insulators, then they have a positive influence on their neighbors. To be specific, Jack has a positive influence on Chris with a probability of 60%. Similarly, because she is a gambler, Mary has a negative influence on Tony with a probability of 90%. Moreover, in the community shown in Fig. 1, only Tony has not been influenced by any gambling insulator. Hence, motivated by the aim to alleviate social problems, such as drinking, smoking, and gambling, this work aims to find a Minimum-sized Positive Influential Node Set (MPINS), which positively influences every individual in a social network with no less than a pre-defined threshold of θ.

MPINS can be applied in various ways, such as the following: For example, a community wants to implement a smoking intervention program. To ensure cost-effectiveness and obtain the maximum effect, the community seeks to select a small number of influential individuals in the community who will attend a quit-smoking campaign. The goal is for all other individuals in the community to be positively influenced by the selected users. Constructing an MPINS can help alleviate the aforementioned social problem, and promote new products in the social network. The following scenario is presented as another motivation example: A small company wants to market a new product in a community. To ensure cost-effectiveness and obtain maximum profits, the company wants to distribute sample products to a small number of initially chosen influential users in the community. The expectation of this company is that these initial users will like the product and positively influence their friends in the community to purchase the product. The goal is to have other users in the community be positively influenced eventually by no fewer than θ of the individuals in the community. In sum, we investigate the following specific problem: Given a social network and a threshold of θ, a minimum-sized subset of individuals in the network is identified such that the subset can result in a positive influence on fewer than θ individuals in the network.

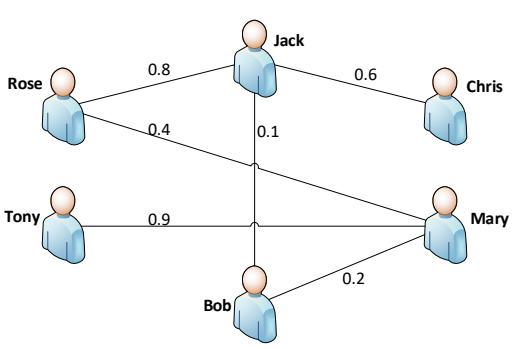


Fig. 1 A social network with social influences on edges.

A related work[7] found a minimum-sized Positive Influence Dominating Set (PIDS), D, so that every other node has at least half of its neighbors in D. In that work, only the positive influence from neighbors is considered, and while the negative influence is ignored. Moreover, the authors in Ref. [7] studied the PIDS selection problem under the deterministic linear threshold model, in which the influence from a pair of nodes is represented by a weight and an individual can be positively influenced when the sum of the weights exceeds a pre-determined threshold[8]. Specifically, the authors in Ref. [7] assumed that the influence of a pair of nodes is always 1, and an individual can be positively influenced when at least half of its neighboring nodes are in D. However, the deterministic linear threshold model is unable to comprehensively characterize the social influence between each pair of nodes in an actual social network because, in the physical world, the strength of the social influence between different pairs of nodes may be different and is actually a probabilistic value[9–13]. Hence, we explore the MPINS selection problem under the independent cascade model considering positive and negative influences, where individuals can positively or negatively influence their neighbors with certain probabilities.

In this paper, we first formally define the MPINS problem. Then, we propose a greedy approximation algorithm to solve this problem. The main contributions of this work are summarized as follows:

• Taking positive and negative influences into consideration, we introduce a new optimization problem called the MPINS selection problem for social networks. To address this problem, we aim to identify the minimum-sized set of influential nodes that could positively influence every node in the network with no less than a predefined threshold of θ. We prove that this problem is an APX-hard problem under the independent

cascade model.

• We define a contribution function and propose a greedy approximation algorithm called MPINS-GREEDY to address the MPINS selection problem. We than analyze the correctness of the proposed algorithm.

• We also conduct extensive simulations and experiments to validate our proposed algorithm. Simulation and experiment results show that the proposed greedy algorithm efficiently solves the MPINS selection problem. More importantly, the solutions obtained by the greedy algorithm are close to the optimal solution of MPINS in small-scale networks.

The rest of this paper is organized as follows: In Section 2, we review related literatures and observe any differences. In Section 3, we first introduce the network model and then formally define the MPINS selection problem and prove its APX-hardness. In Section 6, we present the greedy algorithm and theoretical analysis on the correctness of the algorithm. In Section 8, we present the simulation and experimental results to validate our proposed algorithm. Finally in Section 9, we conclude this paper.

2 Related Work

In this section, we first briefly review related works on social influence analysis. Subsequently, we summarize related literatures on the PIDS problem and the influence maximization problem, followed by some remarks.

2.1 Social influence analysis

Influence maximization was initially proposed by Kempe et al.[1] and it aims to select a set of users in a social network to maximize the expected number of influenced users through several information propagation steps[14]. Empirical studies have been performed on influence learning[10,15], algorithm optimization[16–18], scalability promotion[19–21], and influence of group conformity[4,22]. Saito et al.[23] predicted the information diffusion probabilities in social networks under the independent cascade model. They formally defined the likelihood maximization problem and then applied an Expectation-Maximization (EM) algorithm to solve it. Tang et al.[9,24,25] argued that the effect of the social influence from different angles (topics) may be different. Hence, they introduced Topical Affinity Propagation (TAP) to model topic-related social influence on large social networks. Later, Wang et al.[11] proposed a Dynamic Factor Graph (DFG) model to incorporate time information for the analysis of dynamic social influences. Similarly, Goyal et al.[10] studied the problem of learning the influence probabilities from historical node actions.

2.2 Positive influence dominating set problem

Wang et al.[26] first proposed the PIDS problem under the deterministic linear threshold model, which is to find a set of nodes D such that every node in the network has at least half of its neighbor nodes in D. They proposed a selection algorithm and analyzed its performance on a real online social network data set. Subsequently, Wang et al.[7,27] proved that PIDS is APX-hard and proposed two greedy algorithms through approximation ratio analysis.

He et al.[28] proposed a new optimization problem called the Minimum-sized Influential Node Set (MINS) selection problem. In this problem, the goal is to identify the minimum-sized set of influential nodes, such that every node in the network could be influenced by these selected nodes at no less than a preset threshold. However, they completely neglected the existence of negative influences.

2.3 Influence maximization problem

Domingos and Richardson[29,30] were the first to emphasize the node selection problem when propagating information by using social networks. They considered the social relations of individuals and proposed a probabilistic information propagation model for the problem, as well as several heuristic solutions. Subsequently, Kempe et al.[1,31] formulated the influence maximization problem and studied the problem under two different models, i.e., the linear threshold model and the independent cascade model. They proposed greedy algorithms and analyzed their performance ratios, which are $1-\frac{1}{e}$ under both models. To address the scalability problem of the algorithms in Ref. [1, 31], Leskovec et al.[32] presented a "lazy-forward" optimization scheme of selecting initial nodes, which greatly reduced the number of influence spread evaluations. Chen et al.[33,34] presented the problem of computing exact influence in social networks under both models of #P-Hard and they also proposed scalable algorithms under both models, which are much faster than the greedy algorithms in Refs. [1, 31]. Most recently, considering the data from both the cyber-physical world and online social

networks, Refs. [35–37] proposed methods to provide a comprehensive solution to the problem of influence maximization.

On the other hand, Goyal et al.[38] studied the influence maximization problem from a data-based perspective. They introduced a new model called credit distribution, which directly leverages available propagation traces to learn the manner by which influence flows in the network and adopt it to estimate the expected influence spread. Moreover, they showed that the influence maximization problem under the credit distribution model is APX-hard and designed an approximation algorithm. Zou et al.[39] were the first to add the latency constraint to the influence maximization problem under the linear threshold model and called this modified problem the fast information propagation problem. They further proved that the fast information propagation problem is APX-hard in Ref. [40]. Moreover, two heuristic algorithms are given and their performance ratios are analyzed. Zhang et al.[41] departed from the previous studies on social influence maximization or seed minimization because they considered influence coverage with probabilistic guarantees instead of guarantees on expected influence coverage. They proposed a new optimization problem called Seed Minimization with Probabilistic Coverage Guarantee (SM-PCG) in Ref. [41], and they presented comprehensive theoretical analysis and validated the algorithm by showing the experimental results.

2.4 Remarks

All the above mentioned works fall into three categories: understanding the properties and characteristics of social networks, such as exploring social influences; studying the influence maximization problem with or without time constraint, which has gained considerable attention recently, and addressing the PIDS problem. However, all the aforementioned works did not consider negative influence when they modeled social networks. Aside from taking positive and negative influences into consideration, our work is different from the influence maximization problem because we find a minimum-sized set of individuals that guarantees positive influences on every node in the network with no less than a threshold of θ, while the influence maximization problem focuses on choosing a subset of a predefined size k that maximizes the expected number of influenced individuals. Moreover, our work is also different from the PIDS problem. Given that we study the MPINS selection problem under the independent cascade model and take both positive and negative influences into consideration, our problem is more practical. In addition, PIDS is investigated under the deterministic linear threshold model.

3 Problem Definition and Hardness Analysis

In this section, we first introduce the network model. Subsequently, we formally define the MPINS selection problem and provide some remarks on the proposed problem. Finally, we analyze the hardness of the MPINS selection problem.

3.1 Network model

We model a social network by using an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, where $\mathcal{V}$ is the set of n nodes, denoted by u_i, and $0 \leqslant i < n$. i is called the node ID of u_i. An undirected edge $(u_i, u_j) \in \mathcal{E}$ represents a social tie between the pair of nodes. $\mathcal{P}(\mathcal{E}) = \{p_{ij} \mid \text{if } (u_i, u_j) \in \mathcal{E},\ 0 < p_{ij} \leqslant 1,\ \text{else}\ \ p_{ij} = 0\}$, where p_{ij} indicates the social influence between nodes u_i and u_j*. Notably, social influence can be categorized into positive influence and negative influence. For example, for the smoking intervention program, an individual who initially decided to attend the quit-smoking campaign has a positive influence on all its neighbors, whereas smokers have negative influences on their neighbors. Positive influence and negative influence are formally defined by Definitions 5 and 6, shown in Section 3.2. For simplicity, we assume the links are undirected (bidirectional), that is, two linked nodes have the same social influence (i.e., p_{ij} value) on each other.

3.2 Problem definition

The objective of the MPINS selection problem is to identify a subset of influential nodes as the initialized nodes, such that all the other nodes in a social network can be positively influenced by these nodes with no less than a threshold of θ. For convenience, we call the initial nodes that were selected as active nodes, otherwise, inactive nodes. Therefore, determining how to define positive influence is critical to solve the MPINS selection problem. In the following, we first

*This model is reasonable because many empirical studies have analyzed the social influence probabilities between nodes[10, 11, 23].

formally define some terminologies, and then give the definition of the MPINS selection problem.

Definition 1 Positive influential node set ($\mathcal{I}$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, the positive influential node set is a subset $\mathcal{I} \subseteq \mathcal{V}$, such that all the nodes in $\mathcal{I}$ are initially selected to be the active nodes.

Definition 2 Neighboring set ($\mathcal{B}(u_i)$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, $\forall u_i \in \mathcal{V}$, the neighboring set of u_i is defined as $\mathcal{B}(u_i) = \{u_j \mid (u_i, u_j) \in \mathcal{E}, p_{ij} > 0\}$.

Definition 3 Active neighboring set ($\mathcal{A}^{\mathcal{I}}(u_i)$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, $\forall u_i \in \mathcal{V}$, the active neighboring set of u_i is defined as $\mathcal{A}^{\mathcal{I}}(u_i) = \{u_j \mid u_j \in \mathcal{B}(u_i), u_j \in \mathcal{I}\}$.

Definition 4 Non-active neighboring set ($\mathcal{N}^{\mathcal{I}}(u_i)$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, $\forall u_i \in \mathcal{V}$, the non-active neighboring set of u_i is defined as $\mathcal{N}^{\mathcal{I}}(u_i) = \{u_j \mid u_j \in \mathcal{B}(u_i), u_j \notin \mathcal{I}\}$.

Following by Definitions 3 and 4, we know that the set $\mathcal{A}^{\mathcal{I}}(u_i)$ includes all the active neighboring nodes of u_i and the set $\mathcal{N}^{\mathcal{I}}(u_i)$ includes all the non-active neighboring nodes. How those neighboring nodes collaboratively influence each individual is critical to solve the MPINS selection problem. Next, we define other terminologies.

Definition 5 Positive influence ($p_{u_i}(\mathcal{A}^{\mathcal{I}}(u_i))$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, a node $u_i \in \mathcal{V}$, and a positive influential node set $\mathcal{I}$, we define a joint influence probability of $\mathcal{A}^{\mathcal{I}}(u_i)$ on u_i, denoted by $p_{u_i}(\mathcal{A}^{\mathcal{I}}(u_i))$ as $p_{u_i}(\mathcal{A}^{\mathcal{I}}(u_i)) = 1 - \prod_{u_j \in \mathcal{A}^{\mathcal{I}}(u_i)} (1 - p_{ij})$.

Definition 6 Negative influence ($p_{u_i}(\mathcal{N}^{\mathcal{I}}(u_i))$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, a node $u_i \in \mathcal{V}$, and a positive influential node set $\mathcal{I}$, we define a joint influence probability of $\mathcal{N}^{\mathcal{I}}(u_i)$ on u_i, denoted by $p_{u_i}(\mathcal{N}^{\mathcal{I}}(u_i))$ as $p_{u_i}(\mathcal{N}^{\mathcal{I}}(u_i)) = 1 - \prod_{u_j \in \mathcal{N}^{\mathcal{I}}(u_i)} (1 - p_{ij})$.

Definition 7 Ultimate influence ($\varrho^{\mathcal{I}}(u_i)$). For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, a node $u_i \in \mathcal{V}$, and a positive influential node set $\mathcal{I}$, we define an ultimate influence of $\mathcal{B}(u_i)$ on u_i, denoted by $\varrho^{\mathcal{I}}(u_i)$ as $\varrho^{\mathcal{I}}(u_i) = p_{u_i}(\mathcal{A}^{\mathcal{I}}(u_i)) - p_{u_i}(\mathcal{N}^{\mathcal{I}}(u_i))$. Moreover, if $\varrho^{\mathcal{I}}(u_i) < 0$, we set $\varrho^{\mathcal{I}}(u_i) = 0$. If $\varrho^{\mathcal{I}}(u_i) \geqslant \theta$, where $0 < \theta < 1$ is a predefined threshold, then u_i is said to have been *positively influenced*. Otherwise, u_i has not been positively influenced.

Notably, we assume that the ultimate influences of all active nodes are greater than or equal to θ, i.e., $\forall u_i \in \mathcal{I}, \varrho^{\mathcal{I}}(u_i) \geqslant \theta$. Moreover, if $\mathcal{I} = \varnothing$, then $\forall u_i \in \mathcal{V}, \varrho^{\mathcal{I}}(u_i) = 0$. Finally, we can provide the formal definition of the MPINS selection problem.

Definition 8 MPINS. For social network $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, the MPINS selection problem is to find a minimum-sized positive influential node set $\mathcal{I} \subseteq \mathcal{V}$, such that $\forall u_i \in \mathcal{V} \setminus \mathcal{I}$, u_i is positively influenced, i.e., $\varrho^{\mathcal{I}}(u_i) = p_{u_i}(\mathcal{A}^{\mathcal{I}}(u_i)) - p_{u_i}(\mathcal{N}^{\mathcal{I}}(u_i)) \geqslant \theta$, where $0 < \theta < 1$.

3.3 Problem hardness analysis

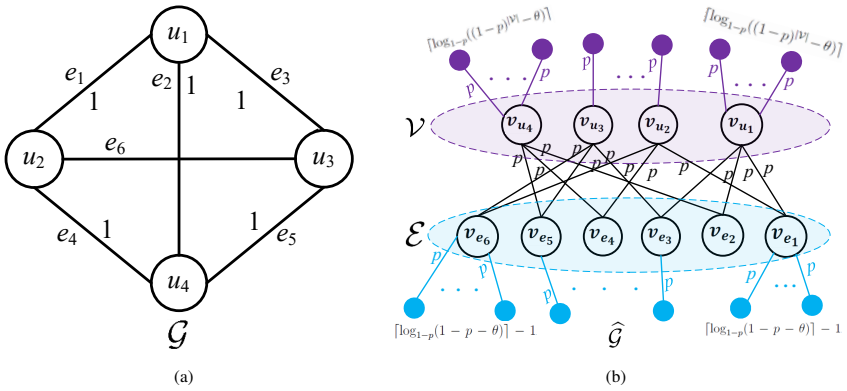
In general, given an arbitrary threshold θ, the MPINS selection problem is APX-hard. We prove the APX-hardness of MPINS by constructing an *L-reduction* from Vertex Cover problem in Cubic Graph (denoted by VCCG) to the MPINS selection problem. The decision problem of VCCG is APX-hard which is proven in Ref. [42]. A cubic graph is a graph where the degree of every vertex is exactly three. Given a cubic graph, VCCG aims to find a minimum-sized vertex cover[†].

First, consider a cubic graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{P}(\mathcal{E}))$, where $\mathcal{P}(\mathcal{E}) = \{1 \mid (u_i, u_j) \in \mathcal{E}; u_i, u_j \in \mathcal{V}\}$, as an instance of VCCG. we construct a new graph $\widehat{\mathcal{G}}$ as follows:

- We create $|\mathcal{V}| + |\mathcal{E}|$ nodes with $|\mathcal{V}|$ nodes $\boldsymbol{v}_{u_i} = \{v_{u_1}, v_{u_2}, \cdots, v_{u_{|\mathcal{V}|}}\}$ representing the nodes in $\mathcal{G}$ and $|\mathcal{E}|$ nodes $\boldsymbol{v}_{e_i} = \{v_{e_1}, v_{e_2}, \cdots, v_{e_{|\mathcal{E}|}}\}$ representing the edges in $\mathcal{G}$.
- We add an edge with influence weight p between nodes v_{u_i} and v_{e_j} if and only if node u_i is an endpoint of edge e_j.
- We attach additional $\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil$ active nodes to each node v_{u_i}, denoted by set $\boldsymbol{v}^{\mathcal{A}}_{u_i} = \{v^j_{u_i} \mid 1 \leqslant j \leqslant \lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil\}$. Obviously, $|\boldsymbol{v}^{\mathcal{A}}_{u_i}| = \lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil$.
- We attach additional $\lceil \log_{1-p}(1-p-\theta) \rceil - 1$ active nodes to each node v_{e_j}, denoted by set $\boldsymbol{v}^{\mathcal{A}}_{e_j} = \{v^j_{e_j} \mid 1 \leqslant j \leqslant \lceil \log_{1-p}(1-p-\theta) \rceil - 1\}$. Obviously, $|\boldsymbol{v}^{\mathcal{A}}_{e_j}| = \lceil \log_{1-p}(1-p-\theta) \rceil - 1$.
- $\widehat{\mathcal{G}} = \{\widehat{\mathcal{V}}, \widehat{\mathcal{E}}\}$, where $\widehat{\mathcal{V}} = \{v_{u_1}, \cdots, v_{u_{|\mathcal{V}|}}\} \cup \{v_{e_1}, \cdots, v_{e_{|\mathcal{E}|}}\} \cup \bigcup_{i=1}^{|\mathcal{V}|} \boldsymbol{v}^{\mathcal{A}}_{u_i} \cup \bigcup_{i=1}^{|\mathcal{E}|} \boldsymbol{v}^{\mathcal{A}}_{e_i}$, $\widehat{\mathcal{E}}$ is the set of all the edges associated with the nodes in $\widehat{\mathcal{V}}$, and $\mathcal{P}(\widehat{\mathcal{E}}) = \{p \mid \text{for every edge in } \widehat{\mathcal{E}}\}$.

With the cubic graph shown in Fig. 2a taken as an example to illustrate the construction procedure from $\mathcal{G}$ to $\widehat{\mathcal{G}}$, four nodes and six edges are found in $\mathcal{G}$. Therefore, we first create $\{v_{u_i}\}_{i=1}^{4}$ and $\{v_{e_j}\}_{j=1}^{6}$ nodes in $\widehat{\mathcal{G}}$. Then

[†]A vertex cover is defined as a subset of nodes in a graph $\mathcal{G}$ such that each edge of the graph is incident to at least one vertex of the set.



(a) (b)

Fig. 2 Illustration of the construction from $\mathcal{G}$ to $\widehat{\mathcal{G}}$.

we add edges with influence weight p between nodes v_{u_i} and v_{e_j} on the basis of the topology shown in $\mathcal{G}$. Subsequently, we add additional $\boldsymbol{v}_{u_i}^{\mathcal{A}} = \{v_{u_i}^j \mid 1 \leqslant j \leqslant \lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil\}$ active nodes to each node v_{u_i} (marked by upper shaded nodes in Fig. 2b. Similarly, we add additional $\boldsymbol{v}_{e_j}^{\mathcal{A}} = \{v_{e_j}^j \mid 1 \leqslant j \leqslant \lceil \log_{1-p}(1 - p - \theta) \rceil - 1\}$ active nodes to each node v_{e_j} (marked by bottom shaded nodes in Fig. 2b. The influence weights on all the additional edges are p. Finally, the new graph $\widehat{\mathcal{G}}$ is constructed as shown in Fig. 2b.

Before we prove that the MPINS selection problem is APX-hard, the following important lemma is introduced.

Lemma 1 $\mathcal{G}$ has a VCCG $\mathcal{D}$ of size at most d if and only if $\widehat{\mathcal{G}}$ has a positive influential node set $\mathcal{I}$ of size at most k by setting $k = |\mathcal{V}|\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil + |\mathcal{E}|(\lceil \log_{1-p}(1 - p - \theta) \rceil - 1) + d$.

4 Proof of Lemma 1

Proof $\Rightarrow$: If $\mathcal{G}$ has a Vertex Cover (VC) $\mathcal{D}$ of size at most d, then we define a set $\mathcal{I}$ in $\widehat{\mathcal{G}}$ consisting of:

- All the additional $|\mathcal{V}|\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil$ active nodes, $\bigcup_{i=1}^{|\mathcal{V}|} \boldsymbol{v}_{u_i}^{\mathcal{A}}$ (marked by upper shaded nodes in Fig. 2b;
- All the additional $|\mathcal{E}|\lceil \log_{1-p}(1 - P - \theta) \rceil - 1$, $\bigcup_{j=1}^{|\mathcal{E}|} \boldsymbol{v}_{e_j}^{\mathcal{A}}$ nodes (marked by bottom shaded nodes in Fig. 2b;
- All the nodes v_{u_i} representing the nodes u_i in the VC $\mathcal{D}$ in $\mathcal{G}$, i.e., $\{v_{u_i} \mid u_i \in \mathcal{D} \text{ in } \mathcal{G}\}$.

Therefore, we have $|\mathcal{I}| = |\mathcal{V}|\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil + |\mathcal{E}|(\lceil \log_{1-p} \theta \rceil - 1) + d \leqslant k$. Now, we need to check whether $\mathcal{I}$ satisfies $\forall v_k \in \widehat{\mathcal{G}}, \varrho^{\mathcal{I}}(v_k) = p_{v_k}(\mathcal{A}^{\mathcal{I}}(v_k)) - p_{v_k}(\mathcal{N}^{\mathcal{I}}(v_k)) \geqslant \theta$.

- For $\forall v_k \in \widehat{\mathcal{G}}$, if $v_k \in \mathcal{I}$, then $\varrho^{\mathcal{I}}(v_k) \geqslant \theta$ according to the assumption.
- For an inactive node $v_{u_i} \in \boldsymbol{v}_{u_i}$, because it connects to $\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil$ active nodes $\boldsymbol{v}_{u_i}^{\mathcal{A}} = \{v_{u_i}^j \mid 1 \leqslant j \leqslant \lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil\}$, we have $\varrho^{\mathcal{I}}(v_{u_i}) = p_{v_{u_i}}(\mathcal{A}^{\mathcal{I}}(v_{u_i})) - p_{v_{u_i}}(\mathcal{N}^{\mathcal{I}}(v_{u_i})) = [1 - (1-p)^{\log_{1-p}((1-p)^{|\mathcal{V}|} - \theta)}] - [1 - (1-p)^{d_i}] \geqslant (1-p)^{d_i} - (1-p)^{|\mathcal{V}|} + \theta \geqslant \theta$, where d_i represents the degree of each node v_{u_i}.
- For every v_{e_j}, it must connect to at most one non-active node and at least one active node $v_{u_i} \in \{v_{u_l} \mid u_l \in \mathcal{D}\}$, which is an active node. Moreover, it also connects to another $\lceil \log_{1-p}(1 - p - \theta) \rceil - 1$ active nodes, $\boldsymbol{v}_{e_j}^{\mathcal{A}} = \{v_{e_j}^j \mid 1 \leqslant j \leqslant \lceil \log_{1-p}(1 - p - \theta) \rceil - 1\}$. Thus, we have $\varrho^{\mathcal{I}}(v_{u_i}) = p_{v_{e_i}}(\mathcal{A}^{\mathcal{I}}(v_{e_i})) - p_{v_{e_i}}(\mathcal{N}^{\mathcal{I}}(v_{e_i})) = [1 - (1-p)^{\lceil \log_{1-p}(1-p-\theta) \rceil - 1 + 1}] - [1 - (1-p)] \geqslant (1-p) - (1-p-\theta) = \theta$.

In summary, if $\mathcal{G}$ has a VC of size d, then $\widehat{\mathcal{G}}$ has a positive influential node set $\mathcal{I}$ with a size of at most k. $\Leftarrow$: Suppose that $\widehat{\mathcal{G}}$ has a positive influential node set $\mathcal{I}$ with a size of at most k. The set $\mathcal{I}$ must include all the nodes in $\bigcup_{i=1}^{|\mathcal{V}|} \boldsymbol{v}_{u_i}^{\mathcal{A}}$, and all the nodes in $\bigcup_{i=1}^{|\mathcal{E}|} \boldsymbol{v}_{e_j}^{\mathcal{A}}$. This result occurs because $\forall v \in (\bigcup_{i=1}^{|\mathcal{V}|} \boldsymbol{v}_{u_i}^{a}) \cup (\bigcup_{i=1}^{|\mathcal{E}|} \boldsymbol{v}_{e_j}^{a})$, v only has one neighbor in $\widehat{\mathcal{G}}$ with the edge between them of influence weight p. Furthermore, $p < \theta$. Therefore, we must include v in $\mathcal{I}$. As a result, without adding more nodes into $\mathcal{I}$, all v_{u_i} nodes are positively influenced already, given that they have $\lceil \log_{1-p}((1-p)^{|\mathcal{V}|} - \theta) \rceil$ active neighbors in $\mathcal{I}$. Now, to make $\mathcal{I}$ a feasible solution of the MPINS selection problem, ensuting that each v_{e_j} either belongs to $\mathcal{I}$ or has at least one neighbor v_{u_i} in $\mathcal{I}$ is sufficient. If v_{e_j} belongs to $\mathcal{I}$, then we may exchange v_{e_j} with its connected v_{u_i}. This approach does not increase the size of set $\mathcal{I}$ and retains the feasibility of the solution. Therefore, we assume that $\mathcal{I}$ does not contain any v_{e_j} so that every v_{e_j} has a neighbor v_{u_i} in $\mathcal{I}$. Given that the current size of set $\mathcal{I}$

is $|\mathcal{V}|\lceil\log_{1-p}((1-p)^{|\mathcal{V}|}-\theta)\rceil+|\mathcal{E}|(\lceil\log_{1-p}(1-p-\theta)\rceil-1)$, the number of v_{u_i} that must be included in $\mathcal{I}$ is d. Let $\mathcal{D}=\{u_i|v_{u_i}\in\mathcal{I}$ in $\widehat{\mathcal{G}}\}$. Consequently, $\mathcal{D}$ is a vertex cover with a size of at most d for $\mathcal{G}$. ■

Next, we prove the complexity of the MPINS selection problem in a general graph in the following theorem.

Theorem 1 The MPINS selection problem is APX-hard.

5 Proof of Theorem 1

Proof An immediate conclusion of Lemma 1 is that $\mathcal{G}$ has a minimum-sized vertex cover of size $\mathrm{OPT}_{\mathrm{VCCG}}(\mathcal{G})$ if and only if $\widehat{\mathcal{G}}$ has a minimum-sized positive influential node set of size

$$\mathrm{OPT}_{\mathrm{MPINS}}(\widehat{\mathcal{G}})=|\mathcal{V}|\lceil\log_{1-p}((1-p)^{|\mathcal{V}|}-\theta)\rceil+|\mathcal{E}|(\lceil\log_{1-p}(1-p-\theta)\rceil-1)+\mathrm{OPT}_{\mathrm{VCCG}}(\mathcal{G}) \quad (1)$$

Note that in a cubic graph $\mathcal{G}$, $|\mathcal{E}|=\dfrac{3|\mathcal{V}|}{2}$. Hence, we have

$$\frac{|\mathcal{V}|}{2}=\frac{|\mathcal{E}|}{3}\leqslant\mathrm{OPT}_{\mathrm{VCCG}}(\mathcal{G}) \quad (2)$$

On the basis of Lemma 1, plugging

$$|\mathcal{V}|=\frac{\mathrm{OPT}_{\mathrm{MPINS}}(\widehat{\mathcal{G}})-\mathrm{OPT}_{\mathrm{VCCG}}(\mathcal{G})}{\lceil\log_{1-p}((1-p)^{|\mathcal{V}|}-\theta)\rceil+\frac{3}{2}(\lceil\log_{1-p}(1-p-\theta)\rceil-1)} \quad (3)$$

into Formula (2), we have

$$\mathrm{OPT}_{\mathrm{MPINS}}(\widehat{\mathcal{G}})\leqslant[2\lceil\log_{1-p}((1-p)^{|\mathcal{V}|}-\theta)\rceil+3\lceil\log_{1-p}(1-p-\theta)\rceil-\frac{1}{2}]\mathrm{OPT}_{\mathrm{VCCG}}(\mathcal{G}) \quad (4)$$

This means that VCCG is L-reducible to MPINS.

In conclusion, we proved that a specific case of the MPINS selection problem is APX-hard, because the VCCG problem is APX-hard. Consequently, the general MPINS selection problem is also at least APX-hard. ■

On the basis of Theorem 1, we conclude that MPINS cannot be solved in polynomial time. Therefore, we propose a greedy algorithm to solve the problem in the next section.

6 Greedy Algorithm and Performance Analysis

MPINS is APX-hard; thus, we propose a greedy algorithm to solve this problem. The proposed algorithm is named MPINS-GREEDY. Before introducing MPINS-GREEDY, we first define a useful contribution function as follows:

Definition 9 Contribution function ($f(\mathcal{I})$). For a social network represented by graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{P}(\mathcal{E}))$, and a positive influential node set $\mathcal{I}$, the contribution function of $\mathcal{I}$ to $\mathcal{G}$ is defined as $f(\mathcal{I})=\sum_{i=1}^{|\mathcal{V}|}\max\{\min(\varrho^{\mathcal{I}}(u_i),\theta),0\}$.

On the basis of the defined contribution function, we propose a heuristic algorithm, which has two phases. First, we find the node u_i with the maximum $f(\mathcal{I})$, where $\mathcal{I}=\{u_i\}$. Afterward, we select a Maximal Independent Set (MIS)[‡] induced by a Breadth-First-Search (BFS) ordering starting from u_i. Second, the pre-selected MIS denoted by $\mathcal{M}$ is used as the initial active node set to perform MPINS-GREEDY, as shown in Algorithm 1. MPINS-GREEDY starts from $\mathcal{I}=\mathcal{M}$. Each time, it adds the node that has the maximum $f(\cdot)$ value into $\mathcal{I}$. The algorithm terminates when $f(\mathcal{I})=|\mathcal{V}|\theta$.

To better understand the proposed algorithm, we use the social network represented by the graph shown in Fig. 3a to illustrate the selection procedure as follows. In the example, $\theta=0.8$. Given that u_1 has the maximum $f(\{u_i\})$ value, we construct a BFS tree rooted at u_1, as shown in Fig. 3b. With the help of the BFS ordering, we find the MIS set which is $\mathcal{M}=\{u_1,u_6\}$. Next, we go to the second phase to perform Algorithm 1. (1) First round: $\mathcal{I}=\mathcal{M}=\{u_1,u_6\}$. (2) Second round: We first compute $f(\mathcal{I}=\{u_1,u_2,u_6\})=4.45$, $f(\mathcal{I}=\{u_1,u_3,u_6\})=3.018$, $f(\mathcal{I}=\{u_1,u_4,u_6\})=3.65$, $f(\mathcal{I}=\{u_1,u_5,u_6\})=3.65$, and $f(\mathcal{I}=\{u_1,u_6,u_7\})=3.778$. Therefore, we have $\mathcal{I}=\{u_1,u_2,u_6\}$, which has the maximum $f(\mathcal{I})$ value. However, $f(\mathcal{I}=\{u_1,u_2,u_6\})=4.45<7\times0.8=5.6$. Consequently,

Algorithm 1 MPINS-GREEDY Algorithm

Require: A social network represented by graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{P}(\mathcal{E}))$; a pre-defined threshold θ.

1: Initialize $\mathcal{I}=\mathcal{M}$
2: **while** $f(\mathcal{I})<|\mathcal{V}|\theta$ **do**
3: choose $u\in\mathcal{V}\setminus\mathcal{I}$ to maximize $f(\mathcal{I}\bigcup\{u\})$
4: $\mathcal{I}=\mathcal{I}\bigcup\{u\}$
5: **end while**
6: **return** $\mathcal{I}$

[‡]MIS can be defined formally as follows: given a graph $G=(V,E)$, an Independent Set (IS) is a subset $I\subset V$ such that for any two vertex $v_1,v_2\in I$, they are not adjacent, i.e., $(v_1,v_2)\notin E$. An IS is called an MIS if we add one more arbitrary node to this subset, the new subset will not be an IS any more.

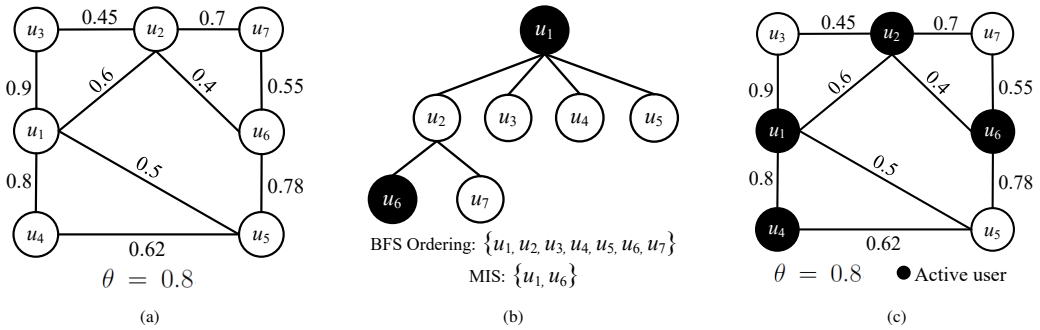


Fig. 3 Illustration of the MPINS-Greedy algorithm.

the selection procedure continues. (3) Third round: We first computer $f(\mathcal{I} = \{u_1, u_2, u_3, u_6\}) = 4.45$, $f(\mathcal{I} = \{u_1, u_2, u_4, u_6\}) = 5.6$, $f(\mathcal{I} = \{u_1, u_2, u_5, u_6\}) = 5.6$, and $f(\mathcal{I} = \{u_1, u_2, u_6, u_7\}) = 4.45$. Therefore, we have $\mathcal{I} = \{u_1, u_2, u_4, u_6\}$[§]. Given that $f(\mathcal{I} = \{u_1, u_2, u_4, u_6\}) = 7 \times 0.8 = 5.6$, the algorithm terminates and outputs set $\mathcal{I} = \{u_1, u_2, u_4, u_6\}$ as shown in Fig. 3c, where black nodes represent the selected influential nodes.

We can easily check that u_3, u_5, and u_7 are all positively influenced. Hence, the constructed $\mathcal{I}$ is a feasible solution for the MPINS selection problem.

The proposed algorithm starts searching from an MIS set ($\mathcal{M}$) instead of an empty set, thereby shortening the algorithm convergence time. Next, we theoretically show the correctness of Algorithm 1 in the following theorem.

Theorem 2 Algorithm 1 produces a feasible solution of the MPINS selection problem. To be specific, (1) Algorithm 1 terminates for certain. (2) $f(\mathcal{I}) = |\mathcal{V}|\theta$ if and only if $\mathcal{I}$ is a positive influential node set, such that every node (i.e., $\forall u_i \in \mathcal{V}$) is positively influenced by nodes in $\mathcal{I}$ no less than θ.

7 Proof of Theorem 2

Proof For (1), on the basis of Algorithm 1, in each iteration, only one node is selected to be added into the output set $\mathcal{I}$. In the worst case, all nodes are added into $\mathcal{I}$ in the $|\mathcal{V}|$-th iteration. Then, $f(\mathcal{I}) = f(\mathcal{V}) = |\mathcal{V}|\theta$ and Algorithm 1 terminates and outputs $\mathcal{I} = \mathcal{V}$. Therefore, Algorithm 1 terminates for certain.

For (2), $\Rightarrow$: if $f(\mathcal{I}) = |\mathcal{V}|\theta$, then $\forall u_i \in \mathcal{V}, \varrho^{\mathcal{I}}(u_i) \geqslant \theta$ followed by Definition 9. Therefore, all nodes in the network are positively influenced.

$\Leftarrow$: if $\forall u_i \in \mathcal{V}, \varrho^{\mathcal{I}}(u_i) \geqslant \theta$, then we obtain $\forall u_i \in \mathcal{V}, \min(\varrho^{\mathcal{I}}(u_i), \theta) = \theta$. According to Definition 9,

$$f(\mathcal{I}) = \sum_{i=1}^{|\mathcal{V}|} \max\{\min(\varrho^{\mathcal{I}}(u_i), \theta), 0\} = |\mathcal{V}|\theta.$$

On the basis of the above two aspects, Algorithm 1 must produce a feasible solution for the MPINS selection problem. ■

8 Performance Evaluation

Given that no existing work studies the MPINS selection problem under the independent cascade model, the simulation and experimental results of MPINS-GREEDY (denoted by MPINS) are compared with the most related work[7] denoted by PIDS, and the optimal solution of MPINS, which is obtained by an exhaustive searching, denoted by OPTIMAL. To ensure fairness of comparison, the condition of termination to the algorithm proposed in Ref. [7] is changed to find a PIDS such that every node in the network is positively influenced with no less than the same threshold of θ in MPINS. We use both real world data sets and synthetic data to demonstrate the effectiveness and efficiency of our proposed model and algorithm. All simulations and experiments were performed on a desktop computer equipped with Intel(R) Core(TM) 2 Quad CPU 2.83 GHz and 6 GB RAM.

8.1 Simulation results

8.1.1 Simulation setting

We build our own simulator to generate random graphs based on the random graph model $G(n, p) = \{G \mid G$ has n nodes, and an edge between any pair of nodes is generated with probability $p\}$. For $G = (V, E) \in G(n, p), u_i, u_j \in V$, and $(u_i, u_j) \in E$, the associated social influence $0 < p_{ij} \leqslant 1$ is randomly generated. Notably, social influence can be categorized into positive influence and negative influence. If one node is selected as the active node, then it has a positive influence on all its neighbors. Otherwise, it has

[§]If a tie exists on the $f(\mathcal{I})$ value, we use the node ID to break the tie.

only a negative influence on its neighbors. For each specific setting, 100 instances are generated. The results are the average values of these 100 instances. In the following, we show the simulation results under different scenarios.

8.1.2 Simulation results on random graphs

The objectives of MPINS and PIDS are to minimize the size of the constructed subsets. In this subsection, we check the size of the solutions of MPINS, PIDS, and OPTIMAL under different scenarios in random graphs. In this simulation, we consider the following tunable parameters: the network size n, the possibility to create an edge p in the random graph model $G(n, p)$, and the user pre-defined influence threshold θ. We adopt exhaustive searching to find the OPTIMAL solution of MPINS, thus testing on large scale networks is impractical. Hence, we first run a set of simulations on small-scale networks, with the network size changing from 10 to 20. Results are shown in Fig. 4.

The impacts of n, p, and θ on the size of the solutions of MPINS, PIDS, and OPTIMAL are shown in Figs. 4a, 4b, and 4c, respectively. Figure 4a indicates that the sizes of the solutions of all the three algorithms increase when n increases. The results occurs because more nodes need to be influenced when the network size increases. In addition, for a specific network size, PIDS produces a larger sized solution than MPINS. This condition occurs because MPINS tries to find the most influential Maximal Independent Set (MIS) of the network first and then adds the node that has the largest $f(\cdot)$ value in each iteration, while PIDS gives the node with the largest degree the highest priority instead. However, a large degree does not necessarily imply a high ultimate influence on the individuals in a social network, because some neighbors may have high negative influences on the individuals. Moreover, MPINS selects an MIS first, which avoids the node selection bias in some specific regions so that more nodes need to be added to the subset to influence all the nodes in the whole network. Furthermore, the size of the MPINS solution is very close to the OPTIMAL result. To be specific, on average, MPINS produces 1.07 more nodes than the OPTIMAL solution, while PIDS produces 3.75 more nodes than the OPTIMAL solution. The results imply that the proposed greedy algorithm MPINS-GREEDY can produce a very close approximation solution to the OPTIMAL solution in small-scale networks.

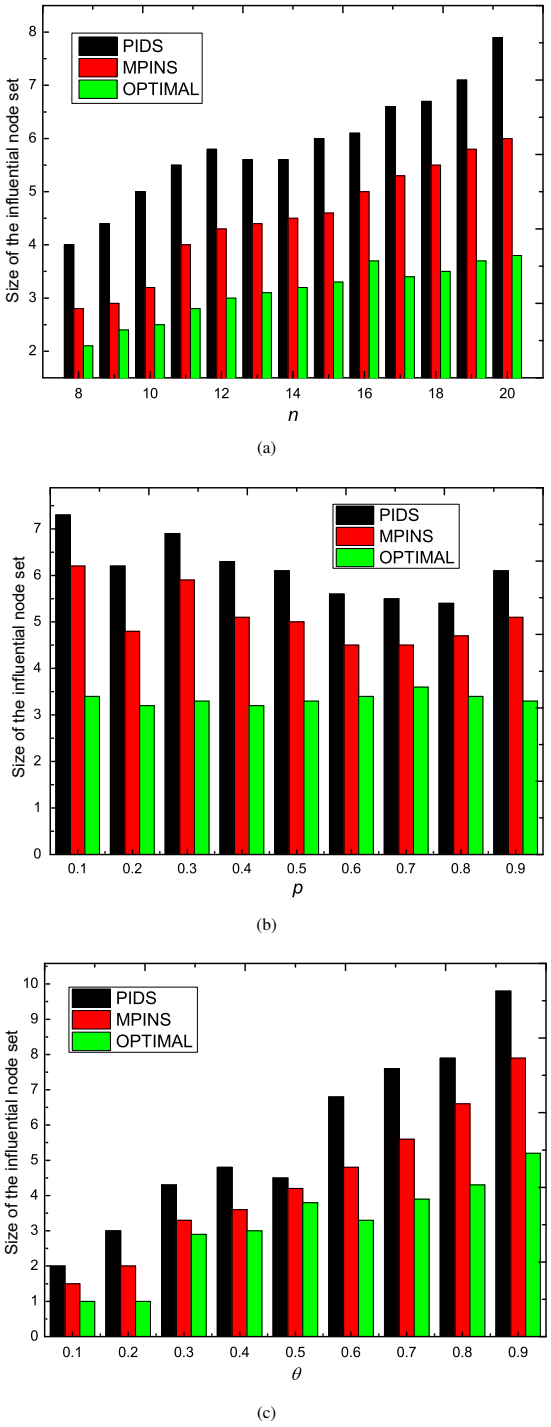


Fig. 4 Size of solutions on small scale networks. The default settings are $n = 15$, $p = 0.5$, and $\theta = 0.5$.

From Fig. 4b, we can see that no obvious trend exists in the solution sizes of all the three algorithms when p increases because the increase in p means more edges in the network so that one specific node may have more negative or positive neighbors. In a very crowded network, distinguishing the pattern of the sizes

of selected influential node sets is difficult. By contrast, for a specific p, PIDS produces larger sized solutions than MPINS because the objective of PIDS is not aimed to obtain the most influential and no-regional-biased nodes in the network. Again, MPINS can construct the solution with a similar size of the OPTIMAL solution. On average, MPINS produces only 1.6 more nodes than the OPTIMAL solution, while PIDS produces 3.16 more nodes than the OPTIMAL solution.

Figure 4c shows that the sizes of all the solutions increase when θ increases, because a large θ value means that more nodes need to be placed in the initial active node set to influence all the other nodes. Furthermore, MPINS has a similar performance with OPTIMAL, and has a better performance than PIDS because the greedy criterion of PIDS is the node with the highest degree first. On average, MPINS produces 1.3 more nodes than the OPTIMAL solutions, while the sizes of PIDS solutions are far from the OPTIMAL results. On average, PIDS produces 3.7 more nodes than the OPTIMAL solution. The reason is the same as we mentioned before.

In addition, we run a set of simulations on medium-scale networks with a network size changing from 100 to 1000. The impacts of n, p, and θ on MPINS and PIDS are shown in Fig. 5. Figure 5a indicates that the solution sizes of MPINS and PIDS increase when n increases because more active influential nodes are needed for larger social networks. Moreover, as n increases, the difference between the sizes of MPINS and PIDS also increases. At a specific n, MPINS can find a positive influential node set that is smaller than that of PIDS because in a small-scale network (i.e., $n < 500$), the initial active node set size is small (no more than 30 from Fig. 5). Hence, the differences between the two methods are not obvious. However, in a medium-scale network, $n = 1000$ for example, our proposed MPINS provides a significant improvement in the size of the initial active node set compared with PIDS. The reason for this scenario is the same as we mentioned earlier. On average, MPINS produces a positive influential node set of size 22.5% less than PIDS.

From Fig. 5b, we can see that the solution sizes of PIDS and MPINS decrease when p increases. p increases, which means the number of edges in the network increases, thereby further implying that the average number of neighbors of each node increases. Hence, one selected active node may influence more

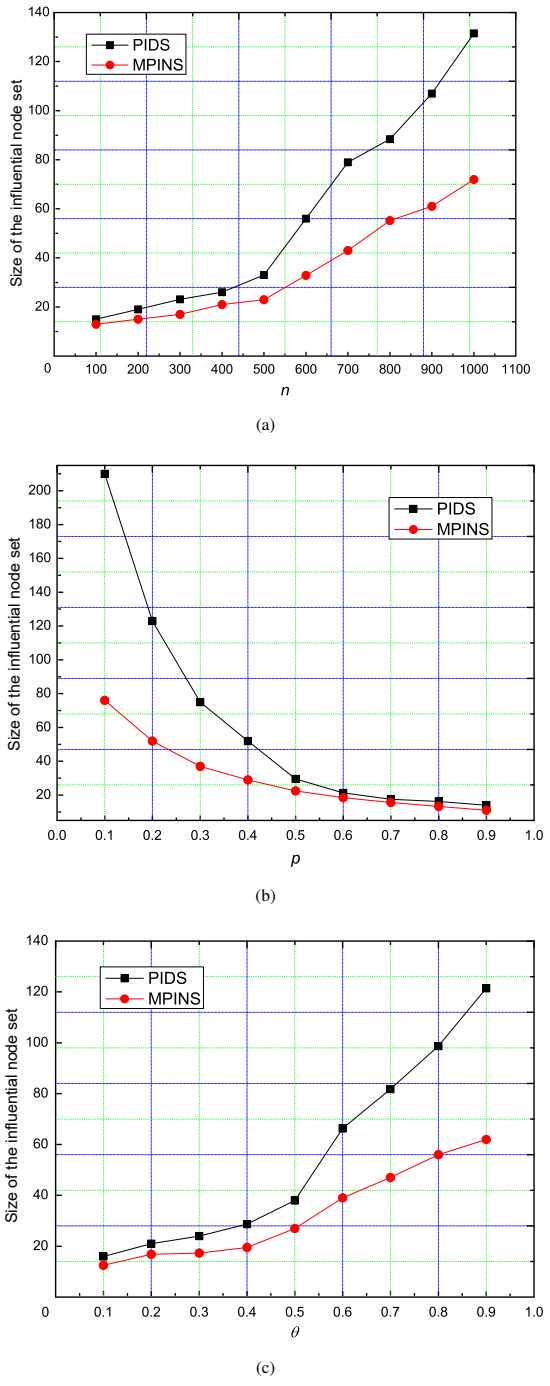


Fig. 5 Size of solutions on large scale networks: The default settings are $n = 15$, $p = 0.5$, and $\theta = 0.5$.

nodes when p increases. For a specific p, PIDS again produces a larger-sized solution than MPINS. When the solution size is small, determining which method outperforms the other is difficult. However, MPINS clearly outperforms PIDS in sparse networks, such as $p = 0.1$. Notably, the decreasing trend of PIDS is very fast when p increases because the degrees of all nodes are small when p is small. Hence, PIDS may find a

solution through many iterations until it finds a solution that ensures that every node in the network is positively influenced by the solution with a threshold of no less than θ. When p is large, larger degree nodes could be added to the solution first so that PIDS might terminate more quickly and is followed by a positive influential node set of a small size. On average, PIDS produces 31.52% more nodes than MPINS.

From Fig. 5c, because of similar reasons analyzed for Fig. 4c, we can see that the solution sizes of solutions of PIDS and MPINS increase when θ increases. Moreover, PIDS outputs an increasing number of nodes than MPINS as θ increases. On average, PIDS produces 23.2% more nodes than MPINS does.

One significant difference between MPINS and PIDS is that MPINS starts the greedy searching on a pre-selected influential MIS set, while PIDS starts searching from an empty set. Moreover, PIDS uses node degree as the greedy search criterion, which might lead to finding some regional-biased nodes so that the final size of the solution may be increased. Our proposed MPINS method selects an MIS first, which avoids the aforementioned dilemma. Figures 6–8 show

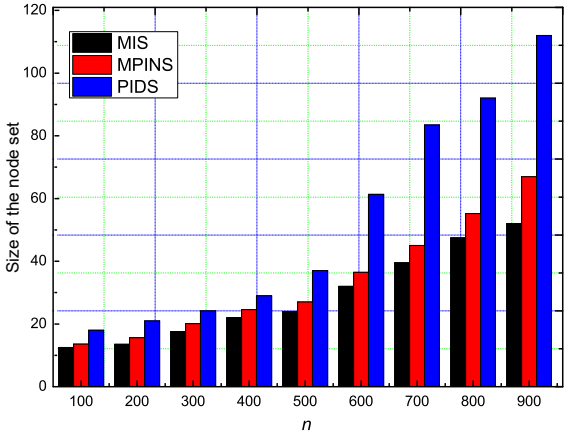


Fig. 6 Size of the node set: The default settings are $p = 0.5$ and $\theta = 0.5$.

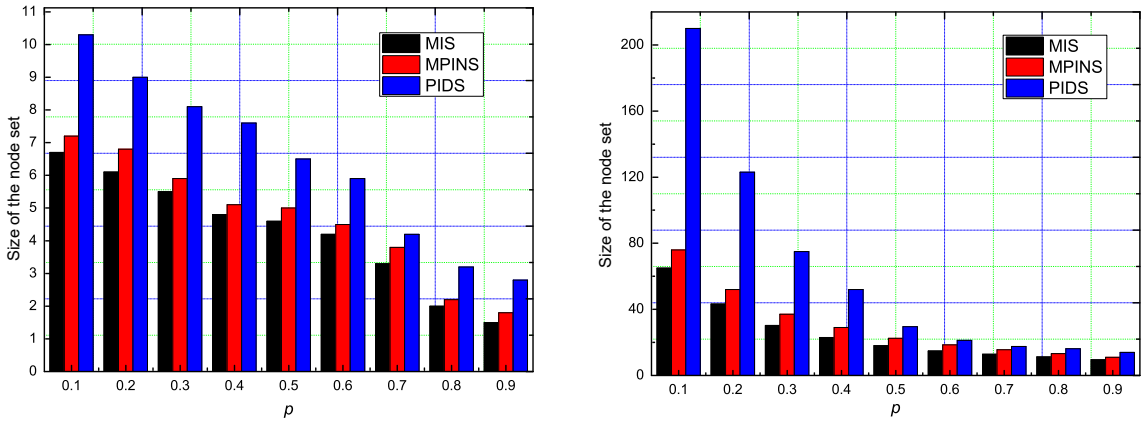


Fig. 7 Size of the node set: (a) n=20 and $\theta = 0.5$; (b) $n = 500$ and $\theta = 0.5$.

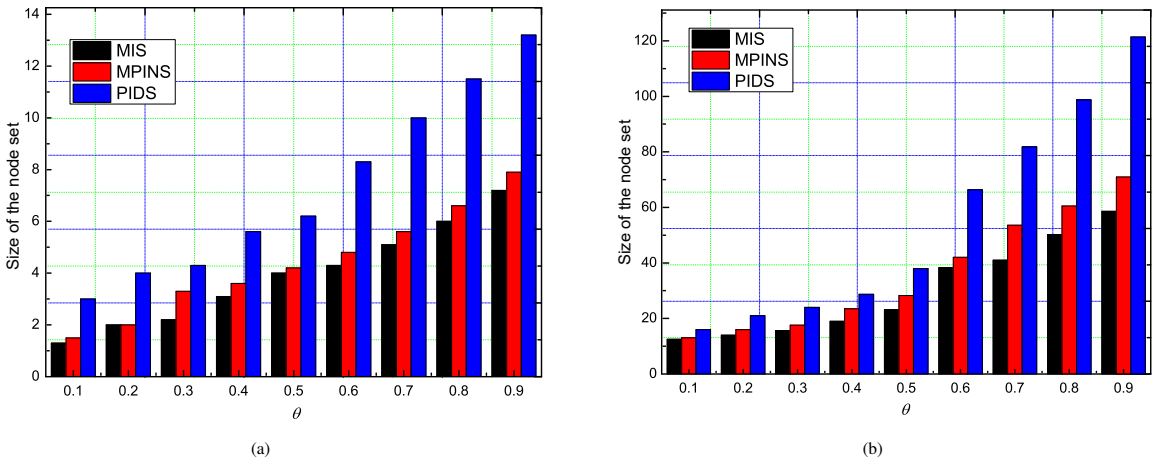


Fig. 8 Size of the node set: (a) $n = 20$ and $p = 0.5$; (b) $n = 500$ and $p = 0.5$.

comparisons of the sizes of MIS, MPINS, and PIDS when n, p, and θ change. These results indicate that only a few iterations of MPINS-GREEDY need to be run to find a solution for MPINS after selecting an influential MIS. However, the iterations for the greedy algorithm proposed for solving PIDS are considerably larger than those of MPINS-GREEDY.

8.1.3 Simulation results on large scale networks

The number of users in social networks has increased explosively. Hence, we run a set of simulations on large-scale networks. The network size changes from 10 000 to 50 000. The impacts of n, p, and θ on MPINS and PIDS are shown in Fig. 9. Figure 9a shows that the solution sizes of MPINS and PIDS both increase when n increases. This increase occurs because more active influential nodes are needed for larger social networks. Moreover, as n increases, the difference between the sizes of MPINS and PIDS also increases. From Fig. 9a, we can clearly see that, in a large-scale network, $n = 50\,000$ for example, our proposed MPINS achieves a significant improvement in the size of the initial active node set compared with PIDS. On average, MPINS produces a positive influential node set with a size 42.1% less than PIDS. From Fig. 9b, because of similar reasons analyzed for Fig. 5, we can see that the solution sizes of PIDS and MPINS increase when θ increases. Moreover, PIDS outputs an increasing number of nodes than MPINS when θ increases. On average, PIDS produces 41.82% more nodes than MPINS does.

From Fig. 9c, we can see that the solution sizes of PIDS and MPINS decrease when p increases. The increase in p means that the number of edges in the network increases, which further implies that the average number of neighbors of each node increases. Hence, one selected active node may influence more nodes when p increases. Similar results can be concluded. (1) For a specific p, PIDS again produces a larger-sized solution than MPINS does. MPINS clearly outperforms PIDS on a very sparse network, such as $p = 0.1$. (2) The decreasing trend of PIDS is very fast when p increases. On average, PIDS produces 34.82% more nodes than MPINS does.

8.2 Experimental results on real data sets

8.2.1 Experimental setting

We also implement experiments run on different kinds of real-world data sets. The first group of data sets,

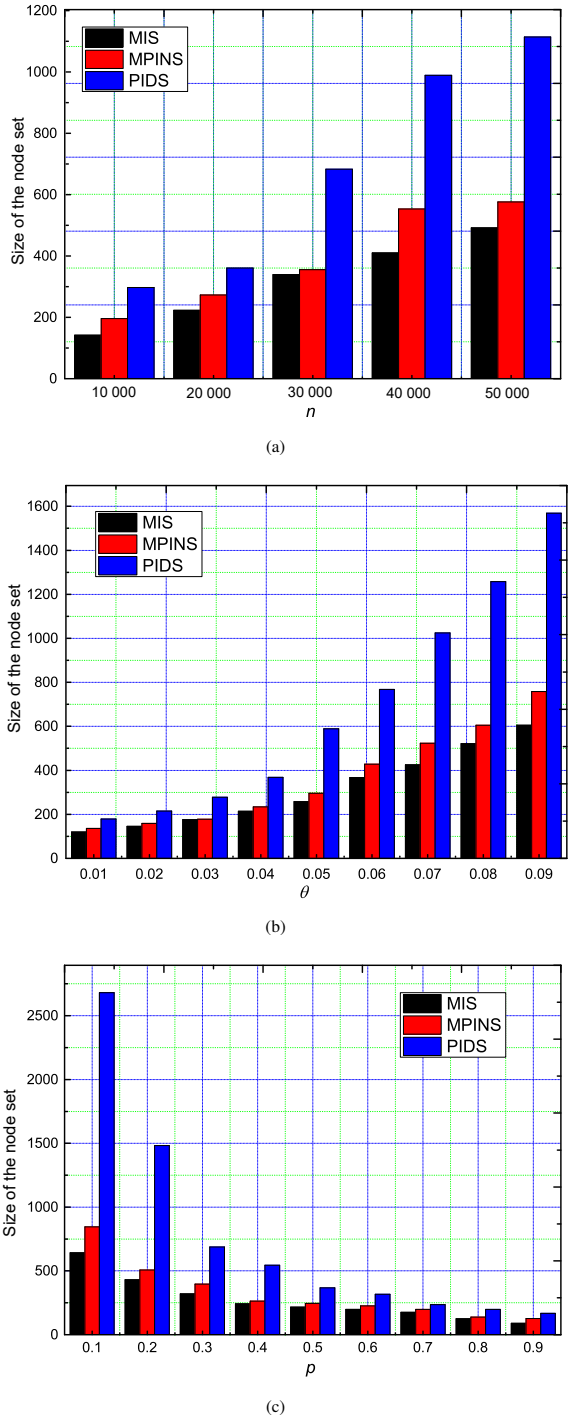


Fig. 9 Size of the node set: (a) $\theta = 0.02$ and $p = 0.2$, (b) $n = 50\,000$ and $p = 0.2$, (c) $\theta = 0.02$ and $n = 50\,000$.

which is shown in Table 1, comes from the Stanford Large Network Dataset Collection (SNAP)[¶], which is a platform for open network data sets collected and maintained by Stanford University. The network

[¶]http://snap.stanford.edu/data/.

Table 1 Data set 1 in our experiment.

Data set	Number of nodes	Number of edges	LWCC(N)	LWCC(E)	LSCC(N)	LSCC(E)	Diameter
A1	262 111	1 234 877	262 111	1 234 877	241 761 (0.922)	1 131 217 (0.916)	29
A2	400 727	3 200 440	400 727	3 200 440	380 167 (0.949)	3 069 889 (0.959)	18
A3	410 236	3 356 824	410 236	3 356 824	390 304 (0.951)	3 255 816 (0.970)	21
A4	403 394	3 387 388	403 364	3 387 224	395 234 (0.980)	3 301 092 (0.975)	21

Note: N stands for nodes, E stands for edges.

statistics are summarized by the number of nodes and edges, the number of nodes and edges in the Largest Weakly Connected Component (LWCC), the number of nodes and edges in the Largest Strongly Connected Component (LSCC), and the diameter (i.e., longest and shortest path). The data collected in Table 1 are based on the Customers Who Bought This Item Also Bought feature of Amazon.com. Four different networks are composed of data collected from March to May in 2003 in Amazon. In each network, for a pair of nodes (products) i and j, an edge exists between them if and only if a product i is frequently co-purchased with product j[43].

Aside from the Amazon product co-purchasing data sets shown in Table 1, we also evaluate our algorithm in the following additional real data sets:

(1) WikiVote: a data set obtained from Ref. [44], which contains the vote history data of Wikipedia[‖]. The data set includes 7115 vertices and 103 689 edges, which contain the voting data of Wikipedia from the inception until January 2008. If user i voted on user j for the administrator election, then an edge will exist between i and j.

(2) Coauthor: a data set obtained from Ref. [45], which holds the coauthors' information maintained by ArnetMiner[**]. We chose the subset that includes 53 442 vertices and 127 968 edges. When the author i has a relationship with author j, one edge will exist between i and j.

(3) Twitter: a data set obtained from Refs. [46, 47], which stores the information collected from Twitter[††]. We selected the subset that includes 92 180 vertices and 188 971 edges, which represent the Twitter account and its follower relationship, respectively.

Moreover, the social influence on each edge (i, j) is calculated by $\frac{1}{\deg(j)}$[48], where $\deg(j)$ is the degree of node j. Similarly, if one node is selected as the active node, then it has a positive influence on all its neighbors. Otherwise, it has only a negative influence on its neighbors.

To better understand the properties of the data in the real-world data sets, Fig. 10 shows the average degree of each data set. Figure 11 summarizes the distribution of the social influence between each pair of nodes in the data sets. Figure 11a shows that most of the edges have social influences that fall in the range $[0.005, 0.05]$ in the Amazon co-purchase data sets (i.e., A1–A4). On the basis of this observation, we let θ change from 0.005 to 0.02 for the Amazon co-purchase data sets in the experiments. Figure 11b shows that most of the edges have social influences that fall in the range $[0.02, 0.10]$ in the WikiVote, Coauthor, and Twitter data sets. Similarly, we let θ change from 0.02 to 0.08 for these three data sets in the experiments.

8.2.2 Experimental results

The impacts of θ on the size of MIS, the solutions of MPINS, and the solution of PIDS on Amazon co-purchase data sets, when θ change from 0.005 to 0.02, are shown in Fig. 12a. As shown in Fig. 12a, the solution sizes of PIDS and MINS increase when θ increases, because when the pre-set threshold becomes large, more influential nodes are required to be chosen to influence the whole network. For one specific θ, MPINS produces smaller influence node sets than PIDS. Moreover, the solution size of MPINS is close to the size of the MIS. Those results are consistent with

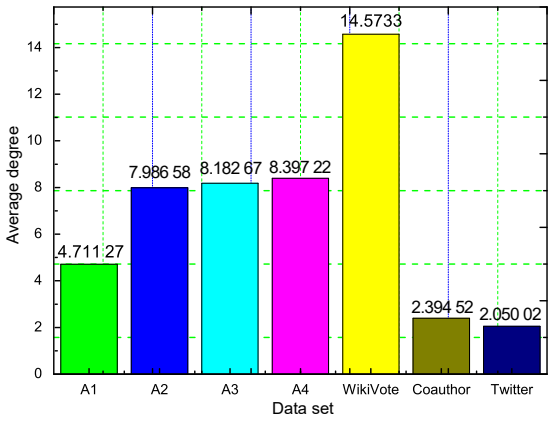


Fig. 10 Average degree of each real-world data sets.

‖http://www.wikipedia.org/.
**http://arnetminer.org, an academic search system.
††https://twitter.com/.

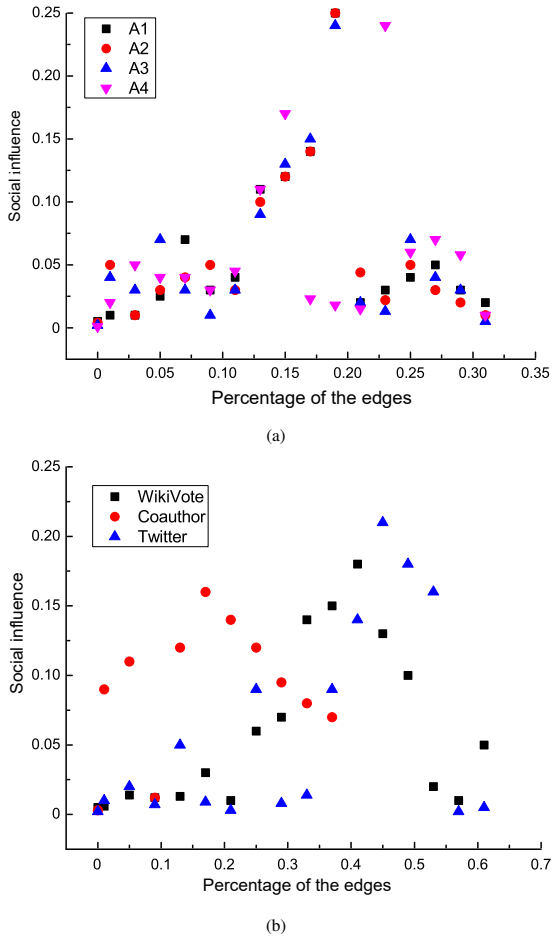


Fig. 11 Probability distribution of (a) Amazon co-purchase data sets and (b) WikiVote, Coauthor, and Twitter data sets.

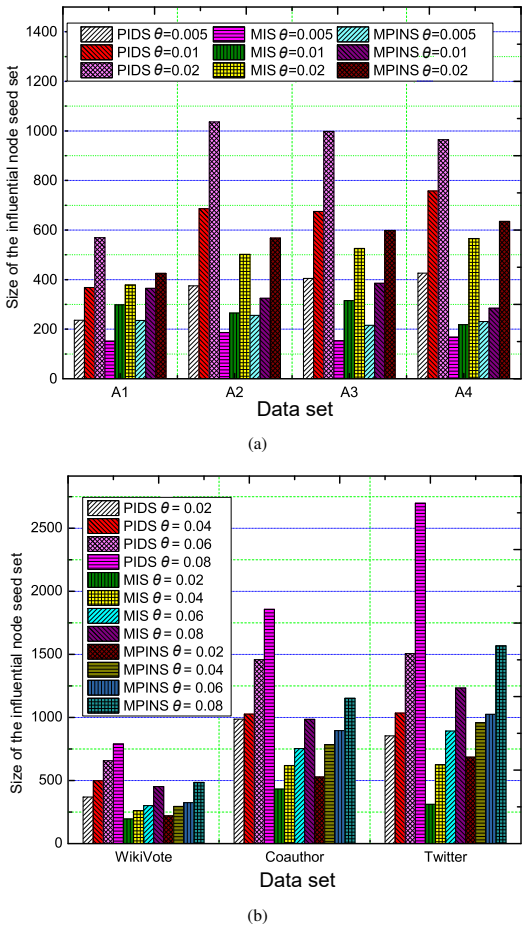


Fig. 12 Size of influential node set in (a) Amazon co-purchase data sets and (b) WikiVote, Coauthor, and Twitter data sets.

the simulation results. For the data set A2, our proposed method MPINS outperforms PIDS significantly. To be specific, MPINS selects 41.31% less influential nodes than PIDS does. On average, the difference between the size of PIDS and MPINS solutions is 37.23%. This result occurs because MPINS chooses the most influential node first instead of the node with the largest degree first. Moreover, the growth rate of the solution size of PIDS is higher than that of MPINS. To be specific, the growth rate of the solution sizes of PIDS is 62.38% on average, while the rate of MPINS is 54.47% on average. Again, the results show that a larger degree does not mean higher influence in a social network. Similarly, the impacts of θ on the size of MIS, the solutions of MPINS, and the solution of PIDS on WikiVote, Coauthor, and Twitter data sets when θ changes from 0.02 to 0.08 are shown in Fig. 12b. As shown in Fig. 12b, the solution sizes of PIDS and MINS increase when θ increases as well. For one specific θ, MPINS produces a smaller influence node set than PIDS. Moreover, the solution size of MPINS is close to the size of the MIS. For the Twitter data set, MPINS outperforms PIDS significantly, i.e., MPINS selects 45.45% less influential nodes than PIDS does. On average, the difference between the sizes of PIDS and MPINS solutions is 36.37%. Moreover, the growth rate of the solution size of PIDS is higher than that of MPINS. To be specific, the growth rate of the solution size of PIDS is 54.1% on average, while that of MPINS is 43.6% on average.

Figure 13 shows how many nodes are selected as the influential nodes represented by the ratio over the total number of nodes in the network. Figure 13a shows the impacts of θ on the ratio of MIS, MPINS, and PIDS on the Amazon co-purchase data sets, while Fig. 13b shows the impacts of θ on the ratio of MIS, MPINS, and PIDS on the WikiVote, Coauthor, and Twitter data sets. We do not repeat the same observed results mentioned earlier. However, one interesting observation here is

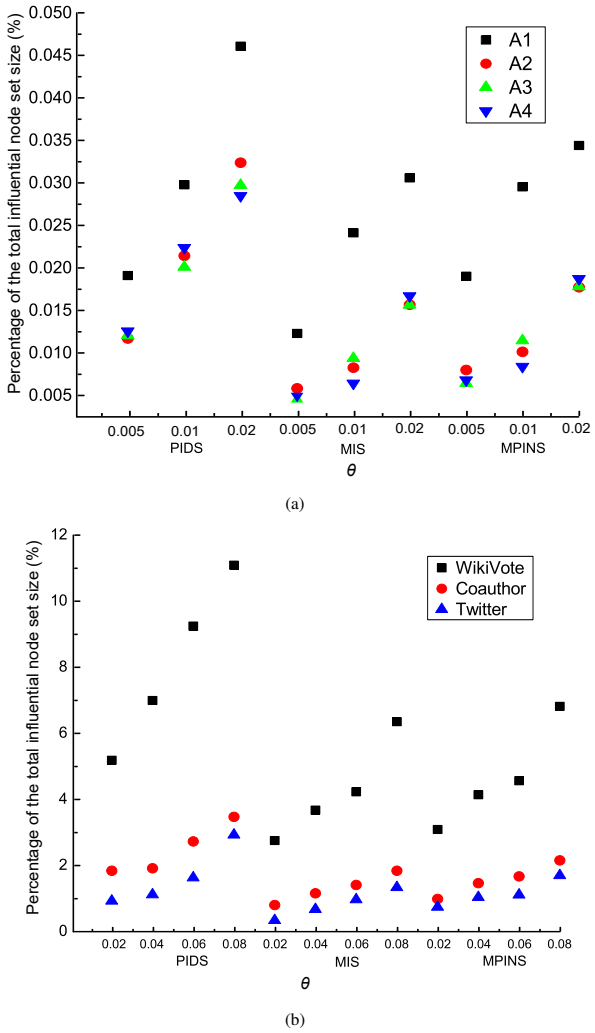


Fig. 13 Percentage of the total influential node set size in (a) Amazon co-purchase data sets and (b) WikiVote, Coauthor, and Twitter data sets.

that much fewer nodes are selected as the influential nodes for Amazon co-purchase data sets compared with the WikiVote, Coauthor, and Twitter data sets. To be specific, in the worst case (for the data set A1 with $\theta = 0.02$), PIDS and MPINS select 0.047% and 0.035% nodes as the influential nodes, respectively. For the WikiVote data set with $\theta = 0.08$, PIDS and MPINS select 11.2% and 7% nodes as the influential nodes, respectively. The results imply that social influences can be more easily propagated in the Amazon co-purchase data sets than in the WikiVote, Coauthor, and Twitter data sets. Amazon usually recommends similar products to users based on users' purchase history, thereby accelerating the influence diffusion process.

Finally, we compare the performance of our proposed method MPINS with that of PIDS and the method denoted by "Random", which randomly chooses a node as the influential node. The impacts of θ on the sizes of the solutions of MPINS, PIDS, and Random when θ changes from 0.02 to 0.08, are shown in Fig. 14 for the WikiVote, Coauthor, and Twitter data sets. As shown in Fig. 14, the solution sizes of Random, PIDS, and MPINS increase when n increases. Moreover, for a specific θ, MPINS produces a smaller influential

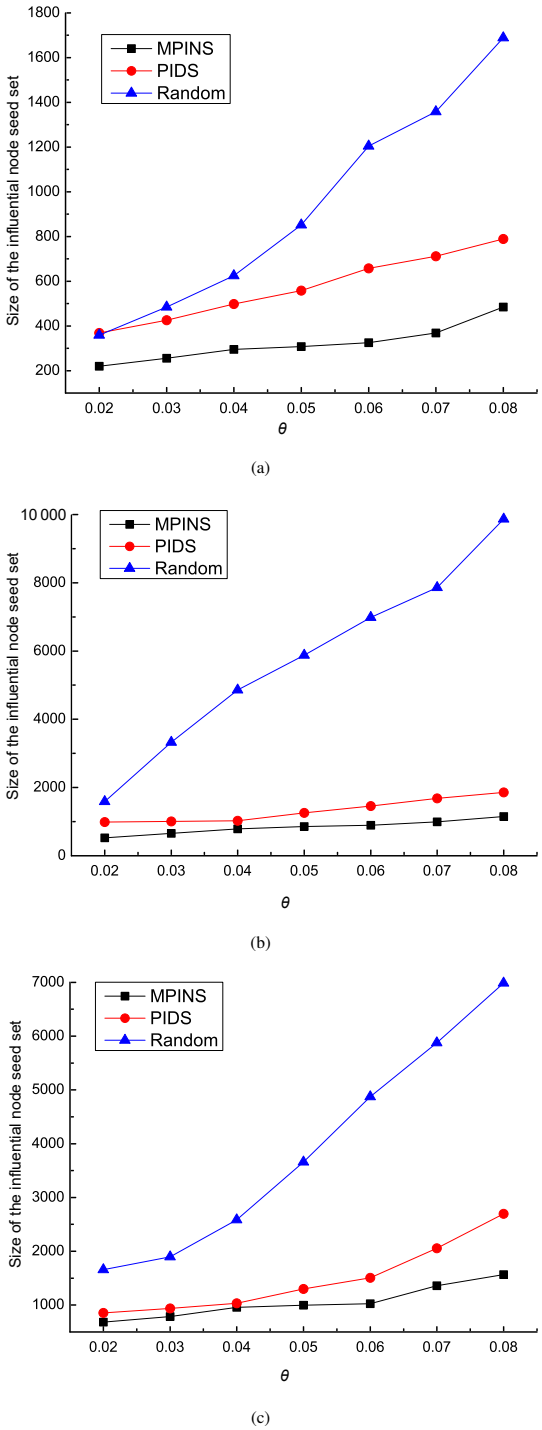


Fig. 14 MPINS vs. PIDS vs. Random in (a) WikiVote data set, (b) Coauthor data set, and (c) Twitter data set.

node set than PIDS. This result is consistent with the simulation results and the previous experimental results. Furthermore, both PIDS and MPINS produce much smaller influential node sets than Random does for a specific θ because Random picks a node randomly without any selection criterion. However, PIDS's selection process is based on degree and our MPINS greedy criterion is based on social influence. Intuitively, both PIDS and MPINS should outperform Random considerably. For the WikiVote data set (shown in Fig. 14a), MPINS selects 48.33% less influential nodes than PIDS does on average. MPINS selects 61.19% less influential nodes than Random does on average. For the Coauthor data set (shown in Fig. 14b), MPINS selects 15.32% less influential nodes than PIDS does on average. MPINS selects 77.13% less influential nodes than Random does on average. For the Twitter data set (shown in Fig. 14c), MPINS selects 23.21% less influential nodes than PIDS does on average. MPINS selects 67.6% less influential nodes than Random does on average.

From simulations on random graphs and experiments on real-world data sets, we can conclude that the constructed initial active node set of MPINS is smaller than that of PIDS. Moreover, the solution of MPINS is very close to the OPTIMAL solutions in small-scale networks.

9 Conclusion

In this paper, we study the MPINS selection problem in social networks, which has useful commercial applications. Through reduction, we show that MPINS is APX-hard under the independent cascade model. Subsequently, a greedy algorithm called MPINS-GREEDY is proposed to solve the problem. We validate our proposed algorithm through simulations on random graphs and experiments on seven different real-world data sets. Simulation and experimental results indicate that MPINS-GREEDY can construct smaller satisfied initial active node sets than the latest related work PIDS. Moreover, for small-scale networks, the performance of MPINS-GREEDY similar to that of the optimal solution of MPINS. Furthermore, MPINS-GREEDY considerably outperforms PIDS in medium- and large-scale networks, sparse networks, and for a high threshold θ.

Acknowledgment

This research was funded in part by the Kennesaw State University College of Science and Mathematics Interdisciplinary Research Opportunities (IDROP) Program, the Provincial Key Research and Development Program of Zhejiang, China (No. 2016C01G2010916), the Fundamental Research Funds for the Central Universities, the Alibaba-Zhejiang University Joint Research Institute for Frontier Technologies (A.Z.F.T.) (No. XT622017000118), and the CCF-Tencent Open Research Fund (No. AGR20160109).

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, Maximizing the spread of influence through a social network, in *Proc. of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 137–146.

[2] K. Saito, M. Kimura, and H. Motoda, Discovering influential nodes for sis models in social networks, in *Proc. International Conference on Discovery Science*, 2009, pp. 302–316.

[3] Y. Li, W. Chen, Y. Wang, and Z. Zhang, Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships, in *Proc. of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 657–666.

[4] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, and J. Yu, Influence maximization by probing partial communities in dynamic online social networks, *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3054, 2016.

[5] X. He, G. Song, W. Chen, and Q. Jiang, Influence blocking maximization in social networks under the competitive linear threshold model, in *Proc. of the 2012 SIAM International Conference on Data Mining*, 2012, pp. 463–474.

[6] W. Lu, F. Bonchi, A. Goyal, and L. V. Lakshmanan, The bang for the buck: Fair competitive viral marketing from the host perspective, in *Proc. of the 19th SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 928–936.

[7] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, and S. Shan, On positive influence dominating sets in social networks, *Theoretical Computer Science*, vol. 412, no. 3, pp. 265–269, 2011.

[8] M. Han and Y. Li, Influence analysis: A survey of the state-of-the-art, in *International Symposium on Bioinformatics Research and Applications*, 2018, pp. 259–264.

[9] J. Tang, J. Sun, C. Wang, and Z. Yang, Social influence analysis in large-scale networks, in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 807–816.

[10] A. Goyal, F. Bonchi, and L. V. Lakshmanan, Learning influence probabilities in social networks, in *Proc. of the Third ACM International Conference on Web Search and Data Mining*, 2010, pp. 241–250.

[11] C. Wang, J. Tang, J. Sun, and J. Han, Dynamic social influence analysis through time-dependent factor graphs,

in *Proc. Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, 2011, pp. 239–246.

[12] M. Han, L. Li, Y. Xie, J. Wang, Z. Duan, J. Li, and M. Yan, Near-complete privacy protection: Cognitive optimal strategy in location-based services, *Procedia Computer Science*, vol. 129, pp. 298–3047, 2018.

[13] L. Liu, M. Han, Y. Zhou, and Y. Wang, LSTM recurrent neural networks for influenza trends prediction, in *International Symposium on Bioinformatics Research and Applications*, 2018, pp. 259–264.

[14] H. Albinali, M. Han, J. Wang, H. Gao, and Y. Li, The roles of social network mavens, in *Proc. of the 12th International Conference on Mobile Ad-hoc and Sensor Networks*, 2016.

[15] M. Han, M. Yan, J. Li, S. Ji, and Y. Li, Generating uncertain networks based on historical network snapshots, in *Proc. COCOON*, 2013, pp.747–758.

[16] A. Goyal, W. Lu, and L. V. Lakshmanan, Celf++: Optimizing the greedy algorithm for influence maximization in social networks, in *Proc. of the 20^{th} International Conference Companion on World Wide Web*, 2011, pp. 47–48.

[17] M. Han, M. Yan, Z. Cai, and Y. Li, An exploration of broader influence maximization in timeliness networks with opportunistic selection, *Journal of Network and Computer Applications*, vol. 63, pp. 39–49, 2016.

[18] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, and N. Lu, A novel evolutionary algorithm based on degree descending search strategy for influence maximization in social networks, *Journal of Network and Computer Applications*, vol. 103, pp. 119–130, 2018.

[19] C. Wang, W. Chen, and Y. Wang, Scalable influence maximization for independent cascade model in largescale social networks, *Data Mining and Knowledge Discovery*, vol. 25, no. 3, p. 545, 2012.

[20] M. Han, Z. Duan, C. Ai, F. W. Lybarger, Y. Li, and A. G. Bourgeois, Time constraint influence maximization algorithm in the age of big data, *International Journal of Computational Science and Engineering*, vol. 15, nos. 3&4, p. 165, 2017.

[21] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, and H. Jin, Scalable influence maximization under independent cascade model, *Journal of Network and Computer Applications*, vol. 86, pp. 15–23, 2017.

[22] J. Tang, S. Wu, and J. Sun, Confluence: Conformity influence in large social networks, in *Proc. of the 19^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 347–355.

[23] K. Saito, R. Nakano, and M. Kimura, Prediction of information diffusion probabilities for independent cascade model, in *Proc. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2008, pp. 67–75.

[24] M. Han, M. Yan, J. Li, S. Ji, and Y. Li, Neighborhood-based uncertainty generation in social networks, *Journal of Combinatorial Optimization*, vol. 28, no. 3, pp. 561–576, 2014.

[25] M. Han, J. Wang, Z. Duan, M. Yan, C. Ai, and Z. Hong, Near-complete privacy protection: Cognitive optimal strategy in location-based services, in *Proc. of the International Conference on Identification, Information and Knowledge in the Internet of Things*. Qufu, China, 2017.

[26] F. Wang, E. Camacho, and K. Xu, Positive influence dominating set in online social networks, in *Proc. International Conference on Combinatorial Optimization and Applications*, 2009, pp. 313–321.

[27] X. Zhu, J. Yu, W. Lee, D. Kim, S. Shan, and D. Z. Du, New dominating sets in social networks, *Journal of Global Optimization*, vol. 48, no. 4, pp. 633–642, 2010.

[28] J. S. He, S. Ji, R. Beyah, and Z. Cai, Minimum-sized influential node set selection for social networks under the independent cascade model, in *Proc. of the 15th ACM International Symposium on Mobile Ad hoc Networking and Computing*, 2014, pp. 93–102.

[29] P. Domingos and M. Richardson, Mining the network value of customers, in *Proc. of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 57–66.

[30] M. Richardson and P. Domingos, Mining knowledgesharing sites for viral marketing, in *Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 61–70.

[31] D. Kempe, J. Kleinberg, and É. Tardos, Influential nodes in a diffusion model for social networks, in *Proc. International Colloquium on Automata, Languages, and Programming*, 2005, pp.1127–1138.

[32] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, Cost-effective outbreak detection in networks, in *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 420–429.

[33] W. Chen, Y. Yuan, and L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in *Proc. Data Mining (ICDM), 2010 IEEE 10th International Conference on*, 2010, pp. 88–97.

[34] W. Chen, C. Wang, and Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 1029–1038.

[35] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, Truthful incentive mechanisms for geographical position conflicting mobile crowd sensing systems, *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2. pp. 324–334, 2018.

[36] M. Han, J. Li, Z. Cai, and Q. Han, Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks, in *Proc. SocialCom 2016*, 2016, pp. 284–292.

[37] M. Han, L. Li, Y. Xie, J. Wang, Z. Duan, J. Li, and M. Yan, Cognitive approach for location privacy protection, *IEEE Access*, vol. 6, pp. 13466–13477, 2018.

[38] A. Goyal, F. Bonchi, and L. V. Lakshmanan, A data based approach to social influence maximization, *Proc. of the VLDB Endowment*, vol. 5, no. 1, pp. 73–84, 2011.

[39] F. Zou, Z. Zhang, and W. Wu, Latency-bounded minimum

influential node selection in social networks, in *Proc. International Conference on Wireless Algorithms, Systems, and Applications*, 2009, pp. 519–526.

[40] F. Zou, J. K. Willson, Z. Zhang, and W. Wu, Fast information propagation in social networks, *Discrete Mathematics, Algorithms and Applications*, vol. 2, no. 1, pp. 125–141, 2010.

[41] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang, Minimizing seed set selection with probabilistic coverage guarantee in a social network, in *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 1306–1315.

[42] D. Z. Du and K. I. Ko, *Theory of Computational Complexity*. John Wiley & Sons, 2011.

[43] J. Leskovec, L. A. Adamic, and B. A. Huberman, The dynamics of viral marketing, *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, p. 5, 2007.

[44] J. Leskovec, D. Huttenlocher, and J. Kleinberg, Predicting positive and negative links in online social networks, in *Proc. of the 19th International Conference on World Wide Web*, 2010, pp. 641–650.

[45] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, Arnetminer: Extraction and mining of academic social networks, in *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 990–998.

[46] J. Hopcroft, T. Lou, and J. Tang, Who will follow you back?: Reciprocal relationship prediction, in *Proc. of the 20th ACM International Conference on Information and Knowledge Management*, 2011, pp. 1137–1146.

[47] T. Lou, J. Tang, J. Hopcroft, Z. Fang, and X. Ding, Learning to predict reciprocity and triadic closure in social networks, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 2, p. 5, 2013.

[48] C. Wang, W. Chen, and Y. Wang, Scalable influence maximization for independent cascade model in largescale social networks, *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 545–576, 2012.

Jing (Selena) He is currently an associate professor in the Department of Computer Science at Kennesaw State University. She received the PhD and MS degrees from Georgia State University (GSU) emphasized in wireless networking. Her research interests include wireless networking and mobile computing, social network analysis, big data analysis on clouds, and Internet of Things. She is a member of IEEE and ACM.

Meng Han currently is an assistant professor in the College of Computing and Software Engineering at Kennesaw State University. He is the Director of Data-driven Intelligence Research (DIR). He got the PhD degree in computer science from Georgia State University. He is currently an ACM member, an IEEE member, and an IEEE COMSOC member. His research interests include data-driven Intelligence, big social data mining, cyber data security & privacy, IoT data, edge data computing, Blockchain technologies, etc.

Tianyu Du is a PhD student in computer science at Zhejiang University. She received the BS degree from Xiamen University. Her research interests include big data driven security and adversarial learning. She is a student member of IEEE and ACM.

Shouling Ji is a ZJU 100-Young Professor in computer science at Zhejiang University and a research faculty in electrical and computer engineering at Georgia Institute of Technology. He received the PhD degree in electrical and computer engineering from Georgia Institute of Technology and the PhD degree in computer science from Georgia State University. His current research interests include big data security and privacy and big data driven security and privacy. He is a member of IEEE and ACM.

Zhao Li is a senior staff scientist and director at Alibaba Group. He received the PhD degree in computer science from University of Vermont. His current research interests include adversarial machine learning and big data driven security.