Efficient targeted influence minimization in big social networks

Wang, Xinjue; Deng, Ke; Li, Jianxing; Xu Yu, Jeffery; Jensen, Christian S.; Yang, Xiaochun

Published in:
World Wide Web

DOI (link to publication from Publisher):
10.1007/s11280-019-00748-z

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Wang, X., Deng, K., Li, J., Xu Yu, J., Jensen, C. S., & Yang, X. (2020). Efficient targeted influence minimization in big social networks. *World Wide Web*, *23*(4), 2323-2340. https://doi.org/10.1007/s11280-019-00748-z

World Wide Web (2020) 23:2323–2340
https://doi.org/10.1007/s11280-019-00748-z

Efficient targeted influence minimization in big social networks

Xinjue Wang[1] · Ke Deng[1] · Jianxin Li[2] · Jeffery Xu Yu[3] · Christian S. Jensen[4] · Xiaochun Yang[5]

Received: 16 March 2018 / Revised: 3 February 2019 / Accepted: 2 October 2019 /
Published online: 19 March 2020

Abstract

An online social network can be used for the diffusion of malicious information like derogatory rumors, disinformation, hate speech, revenge pornography, etc. This motivates the study of influence minimization that aim to prevent the spread of malicious information. Unlike previous influence minimization work, this study considers the influence minimization in relation to a particular group of social network users, called *targeted influence minimization*. Thus, the objective is to protect a set of users, called *target nodes*, from malicious information originating from another set of users, called *active nodes*. This study also addresses two fundamental, but largely ignored, issues in different influence minimization problems: (**i**) the impact of a budget on the solution; (**ii**) robust sampling. To this end, two scenarios are investigated, namely unconstrained and constrained budget. Given an unconstrained budget, we provide an optimal solution; Given a constrained budget, we show the problem is NP-hard and develop a greedy algorithm with an $(1-\frac{1}{e})$-approximation. More importantly, in order to solve the influence minimization problem in large, real-world social networks, we propose a robust sampling-based solution with a desirable theoretic bound. Extensive experiments using real social network datasets offer insight into the effectiveness and efficiency of the proposed solutions.

Keywords Social Network · Influence Minimization · Targeted Campaigns

1 Introduction

Recent years have witnessed the explosive growth of various social media sites such as online social networks, blogs, microblogs, social news websites and virtual social worlds. Online social networks can be used for the diffusion of not only positive information such as innovations, news, and novel ideas, but also malicious information such as disinformation and hate speech. For example, various social media platforms can be used by radical

✉ Ke Deng
ke.deng@rmit.edu.au

Extended author information available on the last page of the article.

organizations and their supporters for a wide range of purposes including recruitment, propaganda, incitement to commit acts of terrorism, and the dissemination of disinformation for terrorist purposes [4].

Research on maximizing the influence of positive information, called *Influence Maximization*, offers insight to social network users on how to best propagate the awareness of products and services and has attracted substantial attention [2, 6, 10, 15]. Likewise, the problem of reducing the influence of negative information, called *Influence Minimization*, is also attracting attention [7–9, 11, 21]. One line of studies on influence minimization aims to find a certain number of edges in social networks such that by deleting these edges, the influence of any information is minimized at the end of the propagation process, no matter which nodes initially have the information [7–9]. Another line of studies assume that a specific set of nodes initially have some information to be spread, The aim is then to delete a certain number of edges or nodes such that the influence of the information is minimized while considering the topics of the information [21] or considering the spread of counter-information from competitors in the same period of time [11, 16].

Unlike the above works, we propose, define, and solve a new problem of so-called targeted influence minimization. This problem and its solutions are relevant to many applications. As we know, the mainstream online social network such as Twitter and Facebook consists of numerous users who are highly diversified in terms of demographic, income, occupation, personal interest, and etc. The problem of targeted influence minimization is to protect a particular group of users in social networks from the influence of negative information. For example, a government agent may want to shield young social network users from pornography or recruitment to terrorism; or a company may initiate a campaign to protect their customers from defamatory information spread by their competitors.

The targeted influence minimization problem can be briefly described as follows: given a set of source nodes I with information to be spread and a set of target nodes T in a social network, the aim is to find the minimum set of edges under a budget constraint such that deleting these edges minimizes the influence from I to T. The deletion of an edge (u_1, u_2) can be considered as persuading u_1 does not spread any information to u_2, or u_2 does not accept any information from u_1. Note that T may include all nodes other than I in a social network in the extreme case. Suppose a set of nodes I regularly spread information for business B_1. A competitor B_2 may initiate a campaign to prevent such information from a set of target nodes T, such as the customers of B_2. To do that, it needs find a set of edges under the campaign budget such that these edges will not pass any information related to B_1. As a consequence, the influence from I to T can be reduced to the minimum level.

All existing studies on influence minimization simply assume the budget is insufficient and provide a greedy algorithm. However, this assumption is not always true. The sufficient budget means the budget is over a threshold such that the optimization objective cannot be further improved, i.e., information propagation has been blocked completely. We develop an optimal solution to completely block propagated information for the target users if the budget is sufficient. If it is, the problem is to find the minimal set of edges which, if deleted, the influence from I to T is completely blocked. Otherwise, the problem is proved to be NP-hard, and a greedy algorithm is developed. To meet the time requirement in handling large social network data, a novel sampling based solution is provided. The contribution of this paper is threefold.

- This work formally defines the targeted influence minimization. It fills the gap of current research in the field of influence minimization in social networks.

- This work investigates two practical scenarios of the targeted influence minimization problem regarding budget sufficiency, which is not fully discussed yet.
- This work proposes robust solutions. In particular, the sampling technique is introduced to solve the problem in large social networks.

This paper is a significantly extended version of publication [20]. While most sections have been improved, the new materials focus on the following aspects. First, a complete literature review section is included to justify the unique position of this study in the field of social network influence minimization. Second, the proofs are provided for each of theorems and lemma which are essential components of this study. Third, the experimental study has been enhanced by two additional tests on effectiveness of the proposed solutions. Fourth, the new version explains that the proposed sampling technique can be successfully applied using disk-based solution; it is particular useful for very large social networks which are not always possible to be loaded in main memory.

The rest of paper is organized as follows. We first cover related work in Section 2 and then define the problem of targeted influence minimization in Section 3. We solve the problem when the budget is unconstrained in Section 4 and when it is constrained in Section 5. Section 6 develops an efficient sampling based solution to enable scalability to large social networks. Finally, we evaluate the effectiveness and efficiency of our proposed solutions using real social network data in Section 7 and conclude in Section 8.

2 Related work

This section introduces the related work in influence minimization which has attracted attention of research community in the past decade.

2.1 Overall influence minimization

This line of study aims to find a certain number of edges in a social network such that deleting these edges minimizes the influence of any information [7–9]. No source nodes and targeted nodes are specified.

Kimura et al. [8, 9] have introduced the influence minimization problem. They define the *contamination degree* of a social network as the average influence of some information on each individual node. Given a budget, they aim to find the set of social network edges such that the number of edges does not exceed a budget and, if the selected edges are deleted, the *contamination degree* of the social networks is minimized assuming the *Independent Cascade* (IC) [6] information diffusion model. They propose a greedy algorithm that iteratively selects the next best edge to be removed based on the reduction in the *contamination degree*. To improve the processing efficiency, they estimate the *contamination degree* by adapting the bond percolation method.

In [7], Khalil et al. have defined the *spread susceptibility* of a social network as $\sum_{i\in V} f_i(S)$, where $f_i(S)$ is the number of nodes influenced by node i after deleting the set of edges, S. Given a vector of information propagation probabilities and a positive integer as the budget, they aim to select a set of edges such that the number of edges does not exceed the budget and, if the selected edges are deleted, the *spread susceptibility* is minimized. They propose a greedy algorithm that computes the loss of susceptibility by removing each edge and then deletes the one leading to the maximum loss. This operation is performed iteratively until the budget is exhausted. They cover two information diffusion models, *Independent Cascade* (IC) and *Linear Threshold* (LT) [6]. If LT is applied, $f_i(S)$ is a monotone and supermodular function; but this is not true for IC. If LT is applied, the greedy algorithm is within $(1 - 1/e)$ of the optimal according to [12]. Wang et al. solve the similar problem using IC [19].

2.2 Source influence minimization

This line of study assumes that a specific set of nodes has some information to be spread. The aim is to delete a certain number of edges such that the influence of the information is minimized in the social network. Yao et al. [21] study this problem and Luo et al. [11] and Song et al. [16] study the spread of counter-information from competitors in the same period of time.

In particular, Luo et al. [11] investigate a different influence minimization problem that considers the influences of two opposite campaigns in a given time period. They assume that once a node becomes active to the information from one campaign, it will not change back to be inactive and will not be active to information from another campaign. Given a positive integer as the budget and a set of nodes that are active for campaign A, they aim to select another set of nodes R with cardinality constrained by the budget such that the number of nodes activated by campaign A is minimized. A greedy algorithm is proposed with a new time-aware influence diffusion model called Continuous-Time Multiple Campaign Diffusion Model, by which is adapted from a model introduced by [14]. The greedy algorithm iteratively populates R with the currently best node according to the objective function, i.e., selecting this node for campaign B will reduce the number of nodes activated for campaign A the most. They prove the objective function monotone and submodular. Therefore, the proposed greedy algorithm is $(1 - 1/e)$ of the optimal [12].

Yao et al. [21] study the influence minimization problem under the Topic-aware Independent Cascade (TIC) diffusion model. Given a set of nodes infected by a textual message and a budget, they aim to select a set of uninfected nodes with a cardinality that is within the budget such that if the nodes are deleted, the number of ultimately infected nodes by the message is minimized. Specifically, the probability that passing the message from an infected node a to a uninfected node b considers whether b is interested in the message based on the log of past propagation. They propose to iteratively deletes the node with the current highest score, where the score is defined using either betweenness or out-degree.

2.3 Remarks

With both source and targeted nodes specified, The most relevant work to our targeted influence minimization is [21] where the edges in social networks have varying weights for different textual messages to be spread. When the message is given and the social network is fixed, it is same as a special case of our problem. The following two points make this study different from existing studies. First, the existing studies including [21] assume that the budget is insufficient even though it is not always true. This study addresses this issue when solving the targeted influence minimization. Second, the existing studies including [21] directly apply greedy algorithm which can also be used to solve our problem; however, we observe the greedy algorithm is hard to handle large social networks. This motivates sampling techniques in our solution when solving the targeted influence minimization.

3 Problem definition

A social network is modeled as a directed graph $G = (V, E)$, where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. A set of nodes $I \subseteq V$ are called active nodes and have information to be diffused in the social network. Another set of nodes $T \subseteq V$, $I \cap T = \emptyset$, are called target nodes and are the recipients of interest.

3.1 Diffusion model

We assume the Linear Threshold (LT) diffusion model [6]. Thus, each edge (u, v) comes with a weight $b_{u,v} \in [0, 1]$ to represent the influence u has on v. If a message is from u, the influence of this message on v is added by $b_{u,v}$. If the message is from all neighbors of v, denoted as $Adj(v)$, then influence, $v.inf$ of the message on v is $\sum_{u\in Adj(v)} b_{u,v}$. An activation threshold, $v.\tau$, is associated with v. If $v.inf \geq v.\tau$, v is activated; otherwise, v is not activated.

It has been shown that diffusion in the LT model is equivalent to the process of reachability under random choice of live edges in graph instances [6]. Given a graph $G = (V, E)$, each node $v \in V$ selects at most one of its incoming edges at random, choosing the edge connecting u to v with probability $b_{u,v}$ and not choosing any other edge with probability $1 - \sum_{u\in Adj(v)} b_{u,v}$; the chosen edge is called *live*. After processing each node in V this way, a graph instance G_x containing only the live edges and all the nodes in G is generated. In G_x, suppose a set of nodes I are active initially; an inactive node $u \in V$ ends up as active if and only if G_x contains a path from any node in I to u.

It has been illustrated that the process of LT model is equivalent to that of the reachability under random choice of *live* edges in graph instances [6]. Given a graph $G(V, E)$, each node $v \in V$ selects at most one of its incoming edges at random, choosing the edge connecting u with probability $b_{u,v}$ and not choosing any edge with probability $1 - \sum_{u\in Adj(v)} b_{u,v}$; the chosen edge is called *live*, and other edges are called *blocked*. After processing each node in V in this way, a graph instance G_x consisting of the live edges is generated. Note G_x contains all nodes in G. In the graph instance G_x, suppose a set of nodes I are active initially; an inactive node $u \in V$ ends up active if and only if there is a path from any node in I to u in G_x.

The set of all graph instances that can be generated from G is denoted as χ_G. The influence of I to a set of nodes $T \subseteq (V \setminus I)$ in graph G under the LT diffusion model is defined as follows:

$$\Lambda_G(I, T) = \sum_{G_x \in \chi_G} Prob[G_x] r_{G_x}(I, T), \tag{1}$$

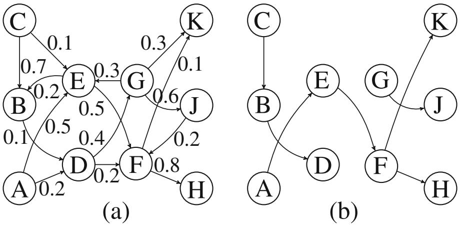
where $r_{G_x}(I, T)$ is the number of nodes in T reachable from any node in I in graph instance G_x, and $Prob[G_x]$ is the probability of graph instance G_x.

Figure 1a illustrates a social network, and an instance graph using the LT diffusion model is shown in Figure 1b. The probability of the instance graph is 0.000504. Suppose $I = \{A, B, C\}$ and $T = \{K, J, H\}$. Then K and H are reachable from A, while J is not reachable from any node in I.

3.2 Targeted influence minimization

A social network $G = (V, E)$ from which a subset of edges $S \subseteq E$ has been deleted is denoted as $G(S)$.

Figure 1 A social network and an instance graph



Definition 1 (Targeted Influence Minimization (TIMin)) Given a social network $G = (V, E)$, a set of active nodes $I \subseteq V$, a set of target nodes $T \subseteq \{V \setminus I\}$ and a positive real number k as a budget, suppose $\mathbb{S} = \{S_1, S_2, \cdots, S_n\}$ contains all possible sets of edges where $|S_i| \leq k$, $1 \leq i \leq n$;

- if there does not exist $S_i \in \mathbb{S}$ such that $\Lambda_{G(S_i)}(I, T) = 0$, TIMin aims to find the set $S_* \in \mathbb{S}$ such that $\Lambda_{G(S_*)}(I, T)$ is minimal;
- if a set $S_i \in \mathbb{S}$ exist such that $\Lambda_{G(S_i)}(I, T) = 0$, TIMin aims to find a set $S_* \in \mathbb{S}$ such that $\Lambda_{G(S_*)}(I, T) = 0$ and $|S_*|$ is minimal.

In the former case, the budget is insufficient to completely block the information propagation from I to T. In the latter case, the budget is sufficient to do so. As an example, consider Figure 1, where $I = \{A, B, C\}$ and $T = \{K, J, H\}$. A budget $k = 2$ is insufficient to completely block the information propagation from I to T. Thus, TIMin aims to find the set of edges S_* such that $\Lambda_{G(S_*)}(I, T)$ is minimized. Given a budget of $k = 10$, there are many sets of edges that, if deleted, will completely block the information propagation from I to T. In this situation, among all such sets of edges, TIMin aims to find one with the minimum number of edges.

Given active nodes I and target nodes T, we initially need to determine whether the budget k is sufficient or not since this is not known in advance. This leads to the following processing framework.

1. The first stage solves the influence minimization with am unconstrained budget, defined as follows.

$$\begin{aligned} \min \quad & |S_i| \\ s.t. \quad & \Lambda_{G(S_i)}(I, T) = 0 \wedge S_i \subset \mathbb{S} \end{aligned} \tag{2}$$

   If $|S_i| \leq k$, the problem is solved by returning S_i because the budget is sufficient to completely block the information propagation from I to T; otherwise, we go to the second stage.
2. The second stage solves the influence minimization with a budget k, defined as follows.

$$\begin{aligned} \min_{S_i} \quad & \Lambda_{G(S_i)}(I, T) \\ s.t. \quad & |S_i| \leq k \wedge S_i \subset \mathbb{S} \end{aligned} \tag{3}$$

4 Budget unconstrained solution

We first examine whether the budget is sufficient to completely block the information propagation from I to T. For this purpose, TIMin with unconstrained budget (i.e., $k = \infty$) is solved as a *minimum cut* or *maximum flow* problem. Let s and t be a source node and sink node in a flow network, respectively. *Maximum flow* problem is to maximize the amount of flow from s to t, i.e., to route as much flow as possible from s to t.

In optimization theory, the *max-flow min-cut theorem* states that the maximum amount of flow passing from the source to the sink is equal to the total weight of the edges in the minimum cut, i.e., equal to the smallest total weight of the edges that, if removed, would disconnect the source from the sink [13]. If multiple sources and multiple sinks exist, the problem is transformed into a single-source and single-sink maximum flow problem by adding two new nodes: one connecting all source nodes and the other connecting all sink nodes; the weights of the new edges connected to the two new nodes are ∞.

Lemma 1 *Given a social network $G = (V, E)$, a set of source nodes $I \subseteq V$, and a set of target nodes $T \subseteq \{V \setminus I\}$, the influence minimization is equivalent to the minimum cut problem if budget $k = \infty$.*

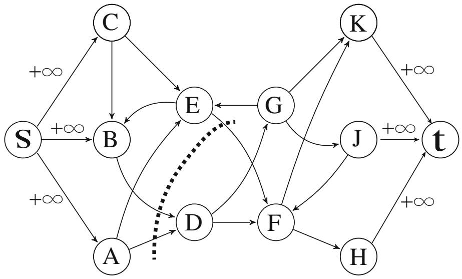
Proof By Definition 1, influence minimization with an unconstrained budget is to identify the minimum set of edges S_* to delete such that by deleting which $\Lambda_{G(S_*)}(I, V) = 0$. As introduced in Section 3, the influence from source nodes is computed using (1). If there is path from any node in I to any node in T, a graph instance exists where $Prob[G_x] > 0$ and $r_{G_x}(I, T) > 0$, and thus $\Lambda(I, V) > 0$. So, $\Lambda(I, V) = 0$ holds only if all possible paths from any node in I to any node in T are blocked. In this situation, no instance graph may have a path from any node in I to any node in T such that $\Lambda_{G(S_*)}(I, V) = 0$.

Therefore, influence minimization with an unconstrained budget is to find a minimum set of edges that, if deleted, disconnect I and T. This is equivalent to the single-source, single-sink minimum cut problem if I and T each contain one node; otherwise, it is equivalent to the multi-source, multi-sink minimum cut problem, which can be transformed into a single-source, single-sink minimum cut problem as discussed above. □

In Figure 2, influence minimization with an unconstrained budget is modeled as a single-source, single-target minimum cut problem. Specifically, a node s is added and linked to all the active nodes in I, and a node t is added and linked to all the target nodes in T. The weight of each edge is infinity.

The *minimum cut* or *maximum flow* problem is well studied [13]. We adopt Dinic's algorithm to solve this problem [3]. Dinic's algorithm uses the concept that a flow is maximum if no path from s to t exists in the residual graph. Given a flow network, if there exists an s-t path, then the algorithm constructs a residual graph based on by applying for the weight

Figure 2 Influence minimization, unconstrained budget



reduction on each edge in the s-t path. The weight to be reduced is the smallest weight on the edges in the path and the updated weights are called *forward capacity*. Meanwhile, the residual graph also records a *backforward capacity* for each edge in the s-t path. The *backforward capacity* for an edge increments by 1 if its *forward capacity* decreases by 1 where the initial *backforward capacity* on each edge is zero.

To improve the efficiency, Dinic's algorithm further proposes the concept of level graph. Each node u in the level graph has an attribute with its shortest distance to s in the residual graph, which maintains information to accelerate the computation of s-t path. If there exists an s-t path left in the residual graph, then it updates the residual graph as well as the level graph. The algorithm stops when no s-t path is left in the residual graph.

The complexity of Dinic's algorithm is $\mathcal{O}(min\{V^{2/3}, E^{1/2}\}E)$ if I and T each contains only one node; otherwise, it is $\mathcal{O}(E^{3/2})$. In Figure 2, the set of edges returned is $S_* = \{(A, D), (B, D), (E, F)\}$ and $|S_*| = 3$. If budget $k \geq |S_*|$, the budget is sufficient to completely block the influence from I to T. The influence minimization is solved and S_* is returned. If the budget $k < |S_*|$, the budget k is insufficient and thus we continue the process in the next section.

5 Budget constrained solution

Theorem 1 *TIMin with an insufficient budget k is NP-hard.*

Proof Maximum coverage problem is known NP-hard [18]. It is a special case of TIMin with an insufficient budget k. Given a number k, elements U, and a number of sets S (the sets may have elements in common), the maximum coverage problem aims to select at most k sets such that the maximum number of unique elements in U are covered, i.e., the cardinality of the union of the selected sets is maximized. In TIMin, a live-path (i.e., a path consisting live edges) from the source nodes I to the target nodes T in instance graphs corresponds to a unique element u in U. An edge e_i in graph $G(V, E)$ correspond to a set S_i in S. The set S_i contains all the live-paths from I to T which, if e is removed, are blocked. Even though it is not true in TMin, let us consider the simplified situation where the corresponding set of each edge is known in advance and all live-paths have the uniform probability. In this simplified situation, TIMin with insufficient budget k aims to select at most k edges which, if deleted, the maximum number of live-paths from I to T are blocked. It is equivalent to the maximum coverage problem. Since TMin is more complex, identifying the best k sets in the maximum coverage problem is a special case of TIMin with insufficient budget k. So, TIMin with budget k is a NP-hard problem □

Due to Theorem 1, we provide a greedy algorithm to solve targeted influence minimization with an insufficient budget.

5.1 Greedy algorithm

The greedy algorithm searches for a set of edges $S \subseteq E$ such that $|S| \leq k$ and the following objective function is maximized.

$$f(S) = \Lambda_G(I, T) - \Lambda_{G(S)}(I, T), \tag{4}$$

where $\Lambda_G(.,.)$ is computed using (1).

The greedy algorithm proceeds iteratively. Initially, S is empty. In each iteration, it computes the value of each edge e in $G(S)$ as follows.

$$value(e) = \Lambda_{G(S)}(I, T) - \Lambda_{G(S')}(I, T), \tag{5}$$

where $S' = S \cup \{e\}$. The value of e, $value(e)$, is the reduction of influence from I to T with and without e in $G(S)$. Among all edges, the one with the maximum value, say e_*, is deleted. Then e_* is inserted into S, and the remaining budget is decremented by 1. The process terminates when the remaining budget reaches 0. The greedy algorithm is an $(1-\frac{1}{e})$-approximation ($\approx$0.632-approximation) since the objective function is non-negative, monotonous, and submodular [7].

6 Sampling-based solution

It is prohibitively expensive to directly generate all graph instances and compute the value of each edge in each iteration. Therefore, we devise a sampling-based solution. The solution is inspired by a recent influence maximization study [17], but significant adaptions are required.

Reverse influence set (RIS) Tang et al. [17] aim to select at most k nodes with maximum influence in a social network. The method is based on RIS that computes the influence of nodes using graph instances. Specifically, the reverse reachable (RR) node set for each node in each graph instance is generated. Given a node v in graph instance G_x, the RR set contains all nodes in G_x that can reach v. Using the sampling method, a number of nodes are randomly selected from V; the RR set for each node is generated using a randomly selected graph instance. So, a number of random RR sets are obtained. If a node u has a great impact on other nodes, u will have high probability of appearing in the random RR sets. As a result, the problem is transformed to the *maximum coverage problem* of identifying at most k nodes that cover the maximum number of the random RR sets. It has been shown that if the number of random RR sets θ is no less than $(8+2\varepsilon)|V|\frac{\ln|V|+\ln\binom{|V|}{k}+\ln 2}{OPT_k\varepsilon^2}$, then RIS returns an $(1-1/e-\epsilon)$-approximate solution with at least $1-|V|^{-1}$ probability ($\epsilon \in (0,1)$) [1].

6.1 Minimum influence path

RIS cannot be applied to our problem without significant modification due to two reasons.

- The random RR set is about node-to-node reachability. In our problem, however, we delete the edges to make reachable-nodes unreachable. While it is straightforward to determine node-to-node reachability, it is more difficult to identify edges the deletion of which makes reachable-nodes unreachable. The reason is that there may be many different paths between two reachable nodes, so deleting an edge does not necessarily block the reachability.
- The random RR set is for the reachability of any node. In our problem, however, only the source nodes I and the target nodes T are relevant.

We propose a novel sampling-based method called *Minimum Influence Path* (MIP) to solve TIMin. The idea is to exploit the fact that each node in a graph instance under the LT diffusion model has at most one incoming edge. Specifically, each node $v \in V$ in the graph instance generation process picks at most one of its incoming edges at random, selecting the edge from $w \in Adj(v)$ with probability $b_{w,v}$, and selecting no edge with probability $1 - \sum_{w \in Adj(v)} b_{w,v}$. Figure 1b shows an example.

As a result, for two nodes v and u, if v is reachable from u in the graph instance, it is easy to observe that the following properties hold: (**i**) there is one and only one path from u to v in the graph instance, and (**ii**) the path is acyclic. Therefore, the information propagation from u to v in this graph instance can be blocked by removing any edge in the path. On the other hand, if v is not reachable from u in the graph instance by deleting an edge e, this does not indicate that v is not reachable from u in other graph instances. However, if v is not reachable from u in many graph instances by deleting e, this implies that the information propagation from v to u is less likely to happen even though it is not impossible. So, the problem is to delete those edges that block the paths from source nodes to target nodes are blocked in many graph instances.

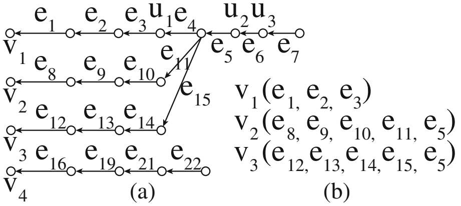
On the other hand, if v is not reachable from u in the graph instance, the information propagation is blocked without deleting any edge. This may occur for two reasons. First, v is not reachable from u in graph G. Second, v is not reachable from u in this graph instance. If v is not reachable from u in many graph instances, this implies that the information propagation from v to u is less likely to happen even though it is not impossible.

Given a node in $v \in T$, the *minimum influence path* in a graph instance is the path to v from any node $u \in I$ with the fewest edges. Figure 3a shows a graph instance where $I = \{u_1, u_2, u_3\}$ and $T = \{v_1, v_2, v_3, v_4\}$. The minimum influence path from I to each target node is shown in Figure 3b. The minimum influence path to v_1 is (e_1, e_2, e_3). Cutting any edge in the minimum influence path will prevent I from reaching v_1 in this graph instance. Intuitively, the edge appearing in more minimum influence paths is more likely to, if deleted, lead to the more influence reduction. In this graph instance, edge e_5 appears in the minimum influence paths of v_2 and v_3 such that deleting e_5 prevents I from reaching two nodes. If deleting e_5 prevents I from reaching many nodes in T in other graph instances, e_5 is likely to be the edge in the solution of MIP.

6.2 Sampling-based greedy algorithm

The pseudo-code of the sampling-based greedy algorithm is presented in Algorithm 1. First, we randomly generate a graph instance in lines 5–7. One node in T is selected randomly in

Figure 3 Reverse influence paths



line 8, and the minimum influence path of this node is generated in line 9. By this way, θ nodes have been sampled, and the minimum influence path is generated for each of them. Note that a graph instance is more likely to be selected if the probability of the graph instance is high. If deleting an edge can prevent I from reaching many nodes in T in many graph instances, this edge is more likely to appear in the minimum influence paths. So, the problem is transformed to the *maximum coverage problem* of selecting at most k edges to cover the sampled nodes as many as possible. In our solution, we assume that the specified budget is sufficient, otherwise, the budget unconstrained solution is applied. To this end, the incremental solution of maximum coverage problem, known as incrementalMC(M), is applied in line 12.

Algorithm 1 Sampling-based solution.

```
   Input: $G = (V, E), I, T, k, \theta$
   Output: $S_*$
1  $i \leftarrow 0$
2  $Mip \leftarrow \emptyset$
3  while $i \leq \theta$ do
4      $j \leftarrow 0$
       // generate a graph instance $i$
5      foreach $v \in V$ do
6          if generateEdge() then
7              randomly select $w \in Adj(v)$ with probability $b_{w,v}$
8      $u \leftarrow$ randomly select a node in $T$
9      $u.Mip \leftarrow$ minInfPath($u$)
10     $Mip \leftarrow Mip \cup u.Mip$
11     $i \leftarrow i + 1$
12 $S_* \leftarrow$ incrementalMC($Mip, k$)
13 return $S_*$
```

The maximum coverage problem is solved using an adapted greedy algorithm that is aware of the budget sufficiency. The pseudo-code is presented in Algorithm 2. The generated minimum influence paths and the corresponding reverse minimum influence paths are used. For each minimum influence path, the algorithm maintains a node $v \in I$ and the list of the edges in the path. For each reverse minimum influence path, it maintains an edge e and a list of the nodes each of which has e in its minimum influence path. The reverse influence minimum paths are constructed while the influence minimum paths are generated (line 2). First, the edge with the longest reverse minimum influence paths is moved to solution S_* (lines 7–8). Then, the nodes in the reverse minimum influence path are processed by finding their minimum influence paths and removing them (line 9); for any edge in the minimum influence paths, its reverse minimum influence paths is found and updated (lines 10–13). The process is repeated until k edges are selected (line 4) or no complete path exists in the remaining influence minimum paths (line 5). The budget sufficiency awareness is implemented by checking whether no complete path exists.

Algorithm 2 Incremental maximum coverage (incrementalMC).

```
Input: $Mip$, $k$
Output: $S_*$
$i = 0$
RMip ← construct reverse minimum influence path with $Mip$
$S_* \leftarrow \emptyset$
while $i \leq k$ do
    if no complete path in $Mip$ then
        break;
    $rp_e$ ← the longest path in RMip
    delete $rp_e$ from RMip
    $S_* \leftarrow S_* \cup e$
    foreach $v \in rp_e$ do
        delete path $p_v$ from $Mip$
        foreach $e \in p_v$ do
            $rp_e$ ← delete $v$ from $rp_e$
return $S_*$
```

Theorem 2 *If $|S_*| \leq k$, the probability that the information propagation from I to T is completely blocked is at most $\frac{1}{n}$; the $|S_*|$ is an $\frac{1}{n}$-approximation of the optimal solution.*

Proof The proof is based on Theorem 3.1 provided by [5]. In the theorem, consider an arbitrary unweighted multigraph $G = (V, E)$ with edge connectivity λ and choose a subset $S \subseteq E$ by indicating each edge $e \in E$ in set S independently with probability p. If $p \geq \frac{20logn}{\lambda}$ then the sampled subgraph $G' = (V, S)$ is connected with probability at least $1 - \frac{1}{n}$.

In this work, each reverse influence path can be modeled as a small graph. Given sets I and T of source and sink nodes, we build a multigraph. As such, the targeted influence minimization from I to T can be transformed into reducing the connectivity of the sampled subgraph G'. Cutting the selected subset S can guarantee that G' is connected with probability at most $\frac{1}{n}$ if $|S| \leq k$. Otherwise, the probability of being disconnected is at most $1 - \frac{k}{|S|}(1 - \frac{1}{n})$. □

6.3 Disk-based path generation

Even though the sampling-based solution is used, it generates θ graph instances. Each graph instance contains all nodes of social network and the selected edges between them. For very large social networks, it is not always possible to load the entire graph instance in main memory. This requires disk-based solution such that the target influence minimization can still be performed. The original social network is stored on disk. The random minimum influence path for each randomly selected node in T is generated without constructing the graph instance. On the disk, the social network is represented as a list of $< v, u >$ for each node v where u is an adjacent node of v. An example for the social network is shown in Figure 4 .

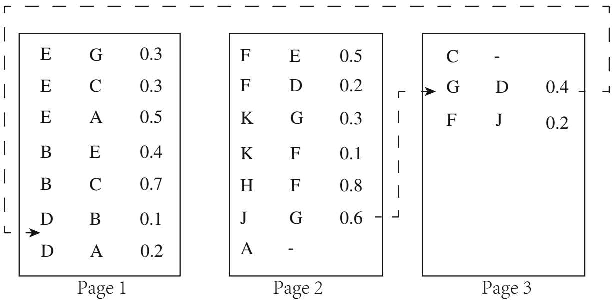


Figure 4 Disk-based solution

Given a randomly selected node $v \in T$, the page containing v is retrieved. A B^+ can be constructed to quickly identify the page ID on which the adjacency information of node v is stored. Once the page is loaded in memory, the adjacent nodes of v are processed. That is, choosing an adjacent node $u \in Adj(v)$ with probability $b_{u,v}$ and not choosing any node with probability $1 - \sum_{u \in Adj(v)} b_{u,v}$. If an adjacent node v' is selected, it is processed in the same way as v; if no adjacent node is selected, the minimum influence path of v is found.

For each adjacent node v' selected, two checks are needed. First, we need determine whether it is the first time that v' is selected in the minimum influence path. If not, selecting v' is illegal since a node selected more than once will lead to cycles in the minimum influence path. Second, we need determine whether v' is in I. If yes, the minimum influence path of v is found; otherwise, the minimum influence path of v grows as discussed. For example in Figure 4, node J is randomly selected. The edge from G to J is chosen, then the edge from D to G is chosen, and finally the edge from D to A is chosen. The minimum influence path of J is $J \leftarrow G \leftarrow D \leftarrow A$.

The total number of edges accessed for generating the minimum influence paths is $\sum_{i=1}^{\theta} l$ where θ is the number of instance graphs and l is the number of edges accessed when generating the minimum influence path in the i^{th} graph instance. In the worst case scenario, when forming each minimum influence path, the entire graph will be accessed. So, the I/O complexity of the algorithm is $\mathcal{O}(\theta n)$ where n is the number of disk pages occupied by the graph G. In practice, some trick can be applied to optimize the I/O. One can try to put adjacent nodes (or edges) in the same page such that, suppose a node v is selected; if retrieving the page where v resides, the edges directly or indirectly pointed to v are more likely to be found in the same page. For example in Figure 4, if $J \leftarrow G$ and $G \leftarrow D$ are in the same page, it is unnecessary to retrieve page 3 for the minimum influence path $J \leftarrow G \leftarrow D \leftarrow A$.

7 Experimental study

We evaluate the effectiveness and efficiency of our proposed algorithms by comparing with two heuristic algorithms called *Random* and *Weight*. *Random* selects edges randomly until budget k is used. *Weight* selects edges with largest edge weights. Their performance are evaluated in different parameter settings using three real-world networks: *Wiki* with 7,115 nodes and 103,689 edges, *Ego-twitter* with 23,370 nodes and 33,101 edges, and *Epinions* with

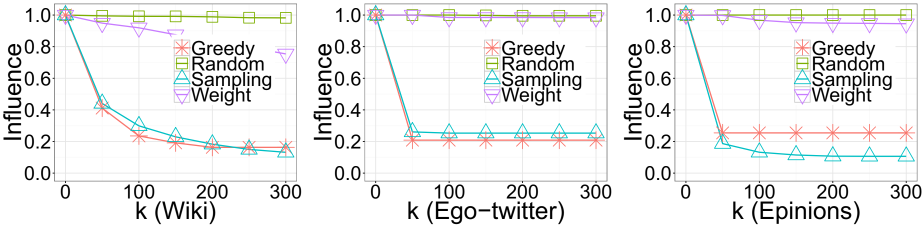


Figure 5 Remaining influence from I on T when varying k

75,879 nodes and 508,837 edges. All the three datasets are downloaded from the *Stanford Dataset Collection.*[1]

7.1 Evaluation of effectiveness

Varying k: Figure 5 shows the experimental results when varying k while the source node set I and the target node set T are fixed in size at 500 unless stated otherwise. The source and target nodes are selected randomly.

The study shows that *Greedy* and *Sampling* are able to greatly reduce the influence of I on T for all three datasets given a sufficiently large value of k. When k is above 100, both solutions are able to reduce the influence by up to 80%. Next, *Random* and *Weight* can slightly reduce the influence in *Wiki*. They do not work for *Ego-twitter* and *Epinions*. *Random* and *Weight* cannot block the influence well because the selection of their deleted edges are not relevant to target users. However, this matter is taken into account in *Greedy* and *Sampling*. So the influences minimized by *Greedy* or *Sampling* are always larger than that of *Random* or *Weight.*

Varying T: We randomly select 500 nodes as the source node set I and set k as 500. Figure 6 shows the results when we increase the target node set from 100 to 500 nodes. *Greedy* and *Sampling* are still able to reduce the remaining influence from I on T by deleting at most 500 edges. *Random* is the worst for all datasets. *Weight* performs better than *Random* in *Wiki* only. The resultant observation is quite interesting. Our proposed solutions *Greedy* and *Sampling* are quite stable at blocking the influence of the source nodes on the target nodes at a certain level.

Varying I: Figure 7 shows the results when we vary the size of the source set I for $k = 500$ and $|T| = 200$. In this study, *Greedy* and *Sampling* can reduce the remaining influence to 0.2 in *Wiki*, which is a dense graph. For *Ego-twitter* and *Epinions*, their performance varies more. Thus, *Greedy* performs better on *Ego-twitter*, and *Sampling* does well on *Epinions*. However, *Random* and *Weight* have the worst performance in all three datasets.

Budget Unconstrained Evaluation: As shown in Figure 8, we can see that the influence from I to T can be blocked completely by deleting a certain number of edges. When $|I| = 500$, $|T| = 100$, it requires 243 edges for *Ego-Twitter*. But more edges must be deleted for *Epinions* and *Wiki* because *Ego-Twitter* dataset is much sparse than *Epinions* or *Wiki* datasets. In order to minimize the influence of I on T in the same parameter settings, it has to delete more edges so that all the paths connecting from I to T can be disconnected. However, when $|T|$ becomes large, it is quite challenging to completely

[1] http://snap.stanford.edu/data/

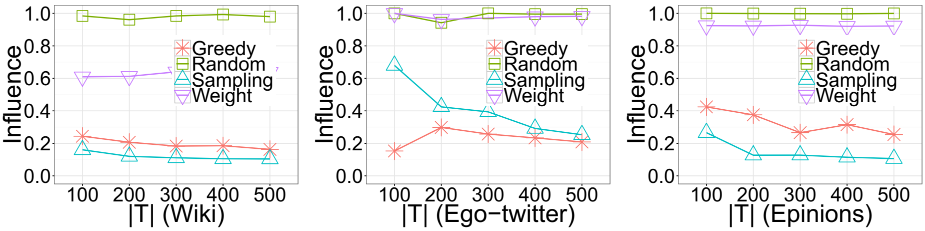


Figure 6 Remaining influence of I on T when varying $|T|$

block the initial users' influence on the target users because a large number of edges need to be deleted. Our sampling solution can be applied to block the majority of the influence.

7.2 Evaluation of efficiency

We evaluate the efficiency of the four solutions when varying k, T, and I. Figures 9, 10 and 11 present the results. Our sampling solution is capable of outperforming the greedy solution by 2 orders of magnitude in all datasets. Both solutions are stable in performance when we increase k. But the time cost of *Greedy* grows with the increase of T or I. Compared with *Greedy* and *Sampling*, *Random* and *Weight* have the best efficiency because their deleted edges can be found without too much computation. But, as we have seen, their lack of effectiveness render them of little use. Therefore, the sampling solution is the best choice for targeted influence minimization in terms of effectiveness and efficiency.

7.3 Evaluation of I/O cost

For very large social networks, it is not always possible to load the entire graph instance in main memory. When the disk-based solution is applied as discussed in Section 6.3, the original social network is stored on disk. In this situation, the I/O cost is tested. We evaluate the number of I/Os when varying T and I on three different datasets as shown in Figures 12. Two situations are compared. One is denoted as $Sampling$ where the edges are randomly stored in different pages, and the other is denoted as $Sampling\_Opt$ which tries to store adjacent nodes in the same page by browsing the graph in width-first manner. In all tests, we suppose no retrieved page will be maintained in memory for reuse.

We observe, $Sampling\_Opt$ outperforms $Sampling$ by 2 orders of magnitude on all datasets and in all settings of I and T. It is interesting to note that in situations of both $Sampling$ and $Sampling\_Opt$ the number of I/O remains the same in large when the size

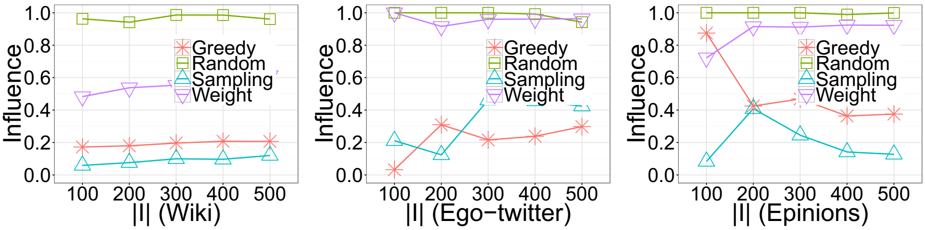


Figure 7 Remaining influence of I on T when varying $|I|$

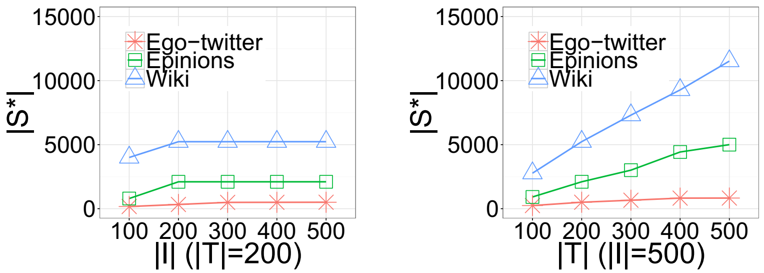


Figure 8 #edges deleted for unconstrained budget

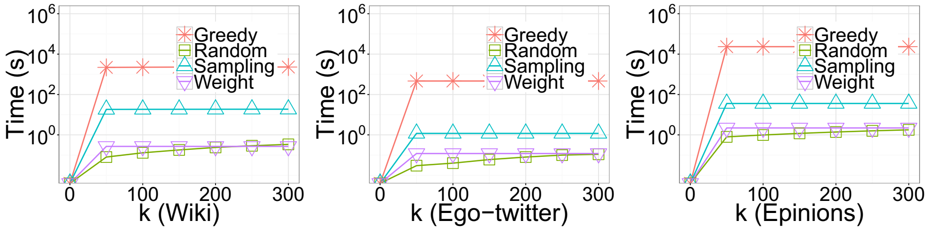


Figure 9 Time cost when varying *k*

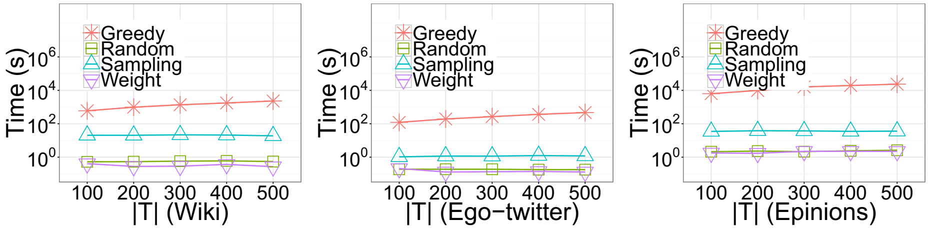


Figure 10 Time cost when varying $|T|$

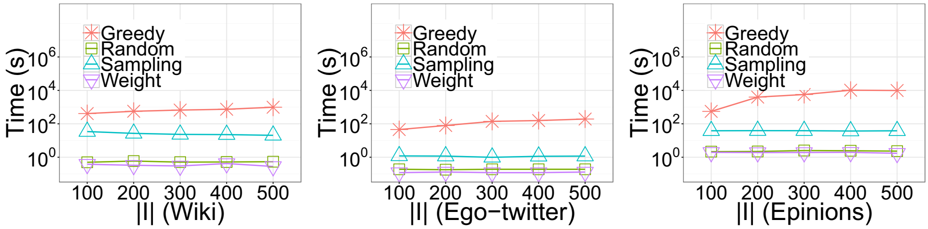


Figure 11 Time cost when varying *I*

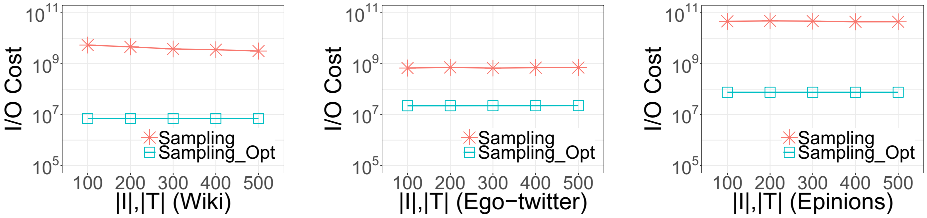


Figure 12 I/O cost when varying both I and T

of I and T increases. As discussed in Section 6.3, it is because that the I/O complexity is determined by θ, the number of instance graphs in the sampling-based algorithm.

At the same time, the increased size of T or I leads to the decrease of length of influence paths and hence the average number of edges accessed tends to decrease when generating the minimum influence paths. Consequently, we can observe a slight but noticeable decrease of I/O cost when the size of T or I increases.

8 Conclusion

In this work, we propose and formalize the problem of targeted influence minimization in social networks that has not previously been studied. We present different solutions that address the computational challenges associated with this problem. We report on empirical studies showing that the proposed solution is capable of quickly blocking 80% or more the influence of source users on target users. The proposed sampling-based solution is efficient when applied to large scale social networks. This is very important because system need to be able to quickly identify the set of edges to be deleted in order to block the source users' influence. A less efficient solution may enable the source users to activate additional users as new source users, who can then spread the malicious information and this way influence the target users. In the future study, we will adapt the techniques developed in this work to information minimization where the temporal factor and the information content are relevant.

Acknowledgements This work is supported by the ARC Discovery Project under grant No. DP160102114.

References

1. Borgs, C., Brautbar, M., Chayes, J.T., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of SODA, pp. 946–957 (2014)
2. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of ACM SIGKDD, pp. 1029–1038 (2010)
3. Dinitz, Y.: Dinitz' algorithm: the original version and even's version. In: Theoretical Computer Science, Essays in Memory of Shimon Even, pp. 218–240 (2006)
4. on Drugs, U.N.O., Crime: The use of the internet for terrorist purposes (2012)
5. Ghaffari, M., Kuhn, F.: Distributed minimum cut approximation. In: Proceedings of DISC, pp. 1–15 (2013)
6. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of ACM SIGKDD, pp. 137–146 (2003)
7. Khalil, E., Dilkina, B., Song, L.: Cuttingedge: Influence minimization in networks. In: Proceedings of Workshop on Frontiers of Network Analysis: Methods, Models, and Applications at NIPS (2013)

8. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of AAAI, pp. 1175–1180 (2008)
9. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of AAAI, pp. 1371–1376 (2007)
10. Li, Y., Zhang, D., Tan, K.: Real-time targeted influence maximization for online advertisements. PVLDB **8**(10), 1070–1081 (2015)
11. Luo, C., Cui, K., Zheng, X., Zeng, D.D.: Time critical disinformation influence minimization in online social networks. In: Proceedings of JISIC, pp. 68–74 (2014)
12. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions—i. Math. Program. **14**(1), 265–294 (1978)
13. Papadimitriou, C.H., Steiglitz, K.: The Max-Flow, Min-Cut theorem. In: Combinatorial Optimization: Algorithms and Complexity, pp. 117–120. Prentice-Hall (1982)
14. Rodriguez, M.G., Leskovec, J., Balduzzi, D., Schölkopf, B.: Uncovering the structure and temporal dynamics of information propagation. Netw. Sci. **2**(01), 26–65 (2014)
15. Shirazipourazad, S., Bogard, B., Vachhani, H., Sen, A., Horn, P.: Influence propagation in adversarial setting: how to defeat competition with least amount of investment. In: Proceedings of ACM SIGMOD, pp. 585–594 (2012)
16. Song, C., Hsu, W., Lee, M.: Temporal influence blocking: Minimizing the effect of misinformation in social networks. In: Proceedings of IEEE ICDE, pp. 847–858 (2017)
17. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of ACM SIGMOD, pp. 75–86 (2014)
18. Vazirani, V.V.: Approximation algorithms. Springer Science & Business Media (2013)
19. Wang, S., Zhao, X., Chen, Y., Li, Z., Zhang, K., Xia, J.: Negative influence minimizing by blocking nodes in social networks. In: Proceedings of Late-Breaking Developments in the Field of Artificial Intelligence, pp. 134–136 (2013)
20. Wang, X., Deng, K., Li, J., Yu, J.X., Jensen, C.S.: Targeted influence minimization in social networks. In: Proceedings of PAKDD (2018)
21. Yao, Q., Shi, R., Zhou, C., Wang, P., Guo, L.: Topic-aware social influence minimization. In: Proceedings of WWW '15 Companion, 1, pp. 139–140 (2015)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Xinjue Wang[1] · Ke Deng[1] · Jianxin Li[2] · Jeffery Xu Yu[3] · Christian S. Jensen[4] · Xiaochun Yang[5]

✉ Jianxin Li
jianxin.li@deakin.edu.au

Xinjue Wang
xinjue.wang@rmit.edu.au

Jeffery Xu Yu
yu@se.cuhk.edu.hk

Christian S. Jensen
csj@cs.aau.dk

Xiaochun Yang
yangxc@mail.neu.edu.cn

[1] RMIT University, Melbourne, Australia

[2] Deakin University, Geelong, Australia

[3] Chinese University of Hong Kong, Sha Tin, China

[4] Aarhus University, Aarhus, Denmark

[5] Northeastern University, Shenyang, China