Limiting the diffusion of information by a selective PageRank-preserving approach

Grigorios Loukides
Cardiff University, UK
gloukides@acm.org

Robert Gwadera*
Cardiff University, UK
gwadera@bluewin.ch

***Abstract*—The problem of limiting the diffusion of information in social networks has received substantial attention. To deal with the problem, existing works aim to prevent the diffusion of information to as many nodes as possible, by deleting a given number of edges. Thus, they assume that the diffusing information can affect all nodes and that the deletion of each edge has the same impact on the information propagation properties of the graph. In this work, we propose an approach which lifts these limiting assumptions. Our approach allows specifying the nodes to which information diffusion should be prevented and their maximum allowable activation probability, and it performs edge deletion while avoiding drastic changes to the ability of the network to propagate information. To realize our approach, we propose a measure that captures changes, caused by deletion, to the PageRank distribution of the graph. Based on the measure, we define the problem of finding an edge subset to delete as an optimization problem. We show that the problem can be modeled as a *Submodular Set Cover* (*SSC*) problem and design an approximation algorithm, based on the well-known approximation algorithm for *SSC*. In addition, we develop an iterative heuristic that has similar effectiveness but is significantly more efficient than our algorithm. Experiments on real and synthetic data show the effectiveness and efficiency of our methods.**

I. Introduction

Controlling the diffusion (propagation) of information in social networks is an important task in multiple domains, such as viral marketing and computer security. In the most common setting, the diffusion starts from a small subset of users who aim to *activate* their friends. The activated friends of these users attempt to activate their own friends, and the diffusion process proceeds similarly until no new users are activated. The diffusing information comes in different forms, such as a link to the website of a new product or to a malicious website to download malware. Typically, the social network is represented as a graph, the initial users correspond to a subset of nodes called *seeds*, and the activation probabilities of nodes are computed according to a diffusion model [13].

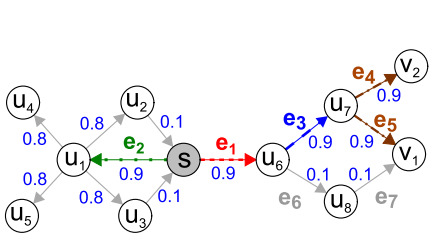
Recently, many works [14], [15], [18], [20] focused on limiting the diffusion of potentially harmful information, by strategically modifying the graph, before the start of the diffusion process. These works aim to find a subset of k edges, whose deletion by a decision maker (*operator*) reduces the expected number of activated nodes at the end of the process (*spread*) as much as possible. However, they consider a rather limited setting, since they assume that: (I) the diffusing information can affect all nodes (i.e., adopt a *collective* approach), and (II) the deletion of each edge has the same impact on the information propagation properties of the graph (i.e., the number of deleted edges determines the ability of the network to propagate information after deletion).

In this work, we consider the problem of limiting information diffusion through edge deletion, in a new setting. Specifically, we propose a *selective* approach that allows specifying the nodes to which information should not be diffused (*vulnerable nodes*) and their maximum allowable activation probability. This flexibility is important in marketing when there are certain classes of users, based on demographics, location, or health condition, that may be harmed by the diffusing information about a product [11], or form and spread negative opinions about it [6]. In addition, our approach determines the impact of deleting an edge subset on the information propagation properties of the graph, using *PageRank* [2], [4], a fundamental model of information propagation based on network topology [1], [22]. This is important because typically there is much other information (i.e., external to the information that is limited), which needs to be propagated on the network after edge deletion. For example, a node with large PageRank score contributes significantly to information propagation. Thus, deleting all its incoming edges, which prevents the propagation of information through the node, should be penalized more heavily than deleting the same total number of edges from many nodes with smaller PageRank.

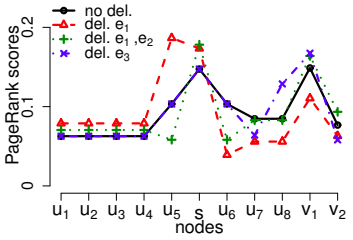
Our approach reduces the activation probability $\mathcal{P}_v$ of each vulnerable node v to at most a threshold $max\mathcal{P}$, while preserving the PageRank distribution of the graph. The activation probabilities are computed by the Linear Threshold (LT) [13] model, a well-established model of the diffusion of potentially harmful information [14], [15]. The threshold $max\mathcal{P}$ is a simple, application-dependent measure of significance (alike the *minimum support* threshold in pattern mining), which models the maximum allowable activation probability for each vulnerable node. The selection of $max\mathcal{P}$ and of vulnerable nodes is performed by the operator, based on domain knowledge (e.g., customer vulnerability analysis and policies [19]).

Since the PageRank score of a node u can be interpreted as the probability that a random walk which starts from a random node ends at u [1], our approach avoids drastic changes to the ability of nodes to propagate any information. Note that this is

*Work done while at EPFL.



(a)



(b)

edge	PRH	$g_{\mathcal{P}_{v_1}}$	$g_{\mathcal{P}_{v_2}}$	$g_{\mathcal{P}_{v_1,v_2}}$
e_1	0.062	0.738	0.729	1.447
e_2	0.062	0	0	0
e_3	0.043	0.729	0.729	1.447
e_4	0.036	0	0.729	0.719
e_5	0.036	0.729	0	0.728
e_6	0.043	0.009	0	0.009
e_7	0.072	0.009	0	0.009

(c)

Algorithm	Del. edges	PRH
AGDE	e_3	0.043
IGDE	$\{e_4, e_5\}$	0.071
[14] $k = 2$	$\{e_1, e_2\}$	0.124

(d)

Fig. 1: (a) Graph and edge probabilities; s is a *seed*, and v_1, v_2 are *vulnerable* nodes. The method of [14] with $k = 1$ (resp., $k = 2$) deletes e_2, (resp., $\{e_1, e_2\}$). (b) The PageRank distribution of the graph in Fig. 1(a) before and after deleting e_1, $\{e_1, e_2\}$, and e_3. (c) PRH, *path probability gain* $g_{\mathcal{P}_{v_1}}$ and $g_{\mathcal{P}_{v_2}}$ used in IGDE, and *aggregate path probability gain* $g_{\mathcal{P}_{v_1,v_2}}$ used in AGDE. (d) The deleted edges and PRH for AGDE, IGDE, and the method of [14] with $k = 2$, when applied to Example 1.

not possible if the LT model, or any other model which only represents a single diffusion process from given seeds, is used instead of PageRank. This is because the use of such a model would allow deleting edges that do not substantially reduce the spread of the diffusion process but harm the ability of the network to propagate other information.

Enforcing our approach is challenging, because: (I) There is an exponential number of edge subsets that can be deleted. (II) There are dependencies between edges, which affect the activation probability of nodes. Specifically, the deletion of an edge (u_l, u) reduces the activation probability of all non-seed nodes reachable from u, because these nodes can no longer be activated by a path that contains (u_l, u). (III) Existing measures [2] that quantify changes to the PageRank distribution cannot be used as optimization criteria in efficient approximation algorithms. In addition, our approach cannot be enforced by existing methods [14], [15] that limit the diffusion of information under the LT model. This is because these methods may not limit the activation probabilities of vulnerable nodes, or they may substantially affect the information propagation on the network, as shown in Example 1.

Example 1. *Consider the graph of Fig. 1(a), where the seed is s, and the vulnerable nodes are v_1 and v_2. The activation probabilities $\mathcal{P}_{v_1}$ and $\mathcal{P}_{v_2}$ in the LT model are equal to* 0.738 *and* 0.729*, respectively. Assume that the activation probabilities $\mathcal{P}_{v_1}$ and $\mathcal{P}_{v_2}$ need to be limited to at most* 0.01*. Applying the method of* [14] *with $k = 1$, deletes $e_2 = (s, u_1)$. This minimizes the expected number of activated nodes. However, $\mathcal{P}_{v_1}$ and $\mathcal{P}_{v_2}$ do not change, since all simple paths from s to v_1 and to v_2 are preserved* [10]*. Using $k = 2$, results in deleting $\{e_1, e_2\}$. This reduces $\mathcal{P}_{v_1}$ and $\mathcal{P}_{v_2}$, to zero. However, the information propagation on the network is significantly affected, since no information can be propagated from u_1, u_2, or u_3 to the nodes on the right of s.*

Our work makes the following contributions:

First, we propose a measure that captures changes, caused by edge deletion, to the PageRank distribution. Our measure, called *PageRank-Harm* (PRH), penalizes the deletion of an edge based on the ratio between the PageRank score and out-degree of the start node of the edge. For example, $e_1 = (s, u_6)$ has a larger PRH than $e_3 = (u_6, u_7)$ in Fig. 1(a), because s has a larger PageRank score than u_6 (see Fig. 1(b)) and s and u_6 have the same out-degree. Since the PageRank score of each node is distributed equally into its out-neighbors, deleting an edge with large PRH incurs a substantial change to the PageRank scores of many other nodes. For instance, deleting e_1 instead of e_3 causes a larger change to the PageRank scores of the nodes in Fig. 1(a), as shown in Fig. 1(b). In addition, we show that the PRH measure can be incorporated into efficient approximation algorithms.

Second, we formally define the optimization problem of *PageRank-preserving Edge Deletion* (*PED*). The problem requires finding an edge subset whose deletion: (I) minimizes changes to the PageRank distribution of the graph according to PRH, and (II) limits the activation probability of each vulnerable node to at most $max\mathcal{P}$. We also prove that *PED* is NP-hard.

Third, we show that *PED*, for a single vulnerable node, can be modeled as a *Submodular Set Cover* (*SSC*) [9], [21] problem. This allows developing an approximation algorithm based on the well-known approximation algorithm for *SSC* [21]. Our algorithm, called GDE, finds an edge subset iteratively. In each iteration, it selects the edge with the minimum ratio between PRH and *path probability gain*, which quantifies the benefit of selecting the edge in terms of decreasing the activation probability $\mathcal{P}_v$ of the vulnerable node. When the deletion of the selected edges can limit $\mathcal{P}_v$ to at most $max\mathcal{P}$, these edges are deleted and the algorithm stops. GDE finds an edge subset whose PRH is larger than that of the optimal solution by at most a logarithmic factor, which depends on the PRH and the path probability gain of the subset.

Fourth, we propose two algorithms for *PED*, when there are multiple vulnerable nodes. The first is an approximation algorithm, called AGDE. The algorithm is similar to GDE, but it selects an edge with small PRH which substantially reduces the activation probabilities of multiple vulnerable nodes simultaneously. Specifically, in each iteration, AGDE selects the edge with the minimum ratio between PRH and benefit in terms of decreasing the activation probability of vulnerable nodes whose activation probability exceeds $max\mathcal{P}$. The benefit is referred to as *aggregate path probability gain*. AGDE achieves a logarithmic approximation ratio, which depends on the PRH and the aggregate path probability gain of the selected edges. Our experiments show that AGDE finds near-optimal solutions (see Fig. 5a). The second algorithm, IGDE, iterates over the vulnerable nodes, in decreasing order

of their activation probability, and applies GDE to approximate the *PED* problem for one vulnerable node per iteration. IGDE is up to two orders of magnitude more efficient than AGDE, because the deleted edges in an iteration are not considered again, and it produces solutions of similar quality, as shown in our experiments. To illustrate AGDE and IGDE, we provide Example 2.

Example 2. AGDE *and* IGDE *were applied to Example 1, using* $max\mathcal{P} = 0.01$. AGDE *selected the edge* e_3 *in Fig. 1(a), which has the minimum ratio between* PRH *and aggregate path probability gain* $g_{\mathcal{P}_{v_1,v_2}}$ *(see Fig. 1(c)). The deletion of* e_3 *limits* $\mathcal{P}_{v_1}$ *and* $\mathcal{P}_{v_2}$ *to at most* 0.01*, thus* AGDE *deleted* e_3. IGDE *considered* v_1 *first, since* $\mathcal{P}_{v_1}$ *is larger than* $\mathcal{P}_{v_2}$*, and selected* e_5*. This is because* e_5 *has the minimum ratio between* PRH *and path probability gain* $g_{\mathcal{P}_{v_1}}$ *among the edges* $\{e_1, e_3, e_5, e_6, e_7\}$*, whose deletion decreases* $\mathcal{P}_{v_1}$ *(see Fig. 1(c)). The deletion of* e_5 *limits* $\mathcal{P}_{v_1}$ *to at most* 0.01*, thus* IGDE *deleted* e_5*. Then,* IGDE *considered* v_2 *and deleted* e_4*. The deletion of* $\{e_4, e_5\}$ *limits both* $\mathcal{P}_{v_1}$ *and* $\mathcal{P}_{v_2}$ *to at most* 0.01*. As shown in Fig. 1(d), the solutions of* IGDE *and the method of* [14] *with* $k = 2$ *have* 65% *and* 186% *larger* PRH *than that of the solution of* AGDE*, respectively.*

II. Background

A. Preliminaries

Let $G(V, E)$ be a directed graph. V is a set of nodes of size $|V|$, and E is a set of edges of size $|E|$. The set of in-neighbors of a node u is denoted with $n^-(u)$ and has size $|n^-(u)|$, which is referred to as the *in-degree* of u. The set of out-neighbors of u is denoted with $n^+(u)$ and has size $|n^+(u)|$, which is referred to as the *out-degree* of u.

A path $q = [(u_1, u_2), \ldots, (u_{m-1}, u)]$ is an ordered set of edges, which has *length* $|q| = m - 1$. A path q in which each node, $u_1, \ldots, u$, is unique (i.e., a path with no cycle) is a *simple* path. A path that starts and ends at the same node is a *cycle* path. We assume simple paths, unless stated otherwise.

To quantify the distance between two probability distributions, $R = \{r_1, \ldots, r_m\}$ and $R' = \{r'_1, \ldots, r'_m\}$, the KL-divergence and the L_1 distance can be used. The L_1 distance quantifies the absolute error between R and R' as $L_1(R, R') = \sum_{i\in[1,m]} |R(r_i) - R'(r'_i)|$, and it is typically used to measure distance between PageRank distributions [2]. The L_1 distance also forms the basis of: (I) the $Gower$ distance, which is defined as $Gower(R, R') = \frac{1}{m} \cdot L_1(R, R')$, and (II) the *Average Relative Error* (ARE), which is defined as $ARE(R, R') = \frac{1}{m} \cdot \sum_{i\in[1,m]} \frac{|R(r_i) - R'(r'_i)|}{R(r_i)}$.

Let U be a universe of elements and 2^U its power set. A set function $f : 2^U \to \mathbb{R}$ is *non-decreasing*, if $f(X) \leq f(Y)$ for all subsets $X \subseteq Y \subseteq U$, *monotone*, if $f(X) \leq f(X \cup u)$ for each $u \notin X$, and *submodular*, if it satisfies the *diminishing returns* property $f(X \cup \{u\}) - f(X) \geq f(Y \cup \{u\}) - f(Y)$, for all $X \subseteq Y \subseteq U$ and any $u \in U \setminus Y$ [16].

B. PageRank

PageRank [4] is a well-established model of information propagation based on network topology [1], [22]. The *PageRank score* of a node u of a graph G is:

$$PR(u, G) = \frac{\alpha}{|V|} + (1 - \alpha) \cdot \sum_{u_l \in n^-(u)} \frac{PR(u_l, G)}{|n^+(u_l)|} \quad (1)$$

where $\alpha \in (0, 1)$ is the *restart probability*, which is usually set to 0.15 [2]. Eq. 1 assumes that each node has out-degree at least 1 (i.e., there are no *dangling* nodes). If there are dangling nodes, we treat them as in [17]. For simplicity of presentation, we henceforth assume that G does not contain dangling nodes. We will write $PR(u)$ for $PR(u, G)$, when G is clear from the context. The *PageRank distribution* of the graph G is denoted with $PR(G)$, and it is defined as the vector of the PageRank scores of all nodes of G [2].

C. The Linear Threshold (LT) model

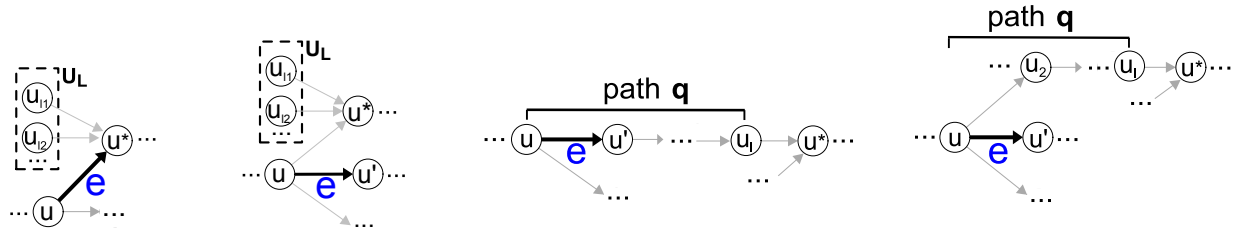
The *edge probability* of an edge (u_l, u) is denoted with $p((u_l, u))$ and reflects how likely u is activated by u_l. For each node u, it holds that $\sum_{u_l \in n^-(u)} p((u_l, u)) \leq 1$. The *path probability* of a path $q = [(u_1, u_2), \ldots, (u_{m-1}, u)]$ is defined as $P(q) = \prod_{e\in q} p(e)$ and reflects how likely u is activated by u_1 through q.

Let $S \subseteq V$ be the set of seeds. Let also $Q_{s,u}$ be the set of paths from a seed s to a non-seed node u of G that do not pass through another seed, and $Q_{S,u} = \cup_{s\in S} Q_{s,u}$. The *activation probability* of u by $Q_{S,u}$ is computed as $\mathcal{P}(u, Q_{S,u}) = \sum_{q\in Q_{S,u}} P(q)$, where $P(q)$ is the path probability of a path q in $Q_{S,u}$ [10]. We denote $\mathcal{P}(u, Q_{S,u})$ with $\mathcal{P}_u$, when $Q_{S,u}$ is clear from the context. We also define the *activation graph* $\tilde{G}_u$ of u as the subgraph of G which is induced by the edges of all paths in $Q_{S,u}$.

The exact computation of $\mathcal{P}_u$ is a $\#P$-hard problem for general graphs [7]. However, the path probability of each path decreases exponentially with the path length. Thus, $\mathcal{P}_u$ can be estimated accurately and efficiently, based on the subset of paths in $Q_{S,u}$ whose seeds are "close" to u [10]. To find these paths, we adapt the depth-first-search-based algorithm of [10]. For each seed, the algorithm iteratively extends each path from the seed and prunes it, if its path probability is lower than a threshold h. Then, $\mathcal{P}_u$ is computed based on the paths from seeds to u that are found by the algorithm, and $\tilde{G}_u$ is constructed as the graph induced by the edges of these paths. The threshold $h \in [0, 1]$ represents the maximum tolerable estimation error and is operator-specified [10]. The impact of h on our approach is studied in Section VIII.

III. The PRH measure

The deletion of an edge affects the PageRank score of the end node of the deleted edge, according to Eq. 1. In addition, the PageRank score of this node is distributed into its out-neighbors. Thus, the PageRank scores of these nodes change, and the change is propagated similarly. Therefore, edge deletion may incur a substantial change to the PageRank distribution. Minimizing the change in our problem is challenging, because: (I) there are $O(2^{|E|})$ edge subsets that can be deleted, and (II) existing measures that capture changes to the PageRank distribution (see Section II-A) are not monotone and cannot be incorporated into efficient approximation algorithms. Therefore, we propose PRH, a monotone measure that



I(a) I(b) II(a) II(b)

Fig. 2: Cases that summarize the relation between $\delta(u^*)$ and $PRH(e)$. The deleted edge e is in bold, U_L is the set of in-neighbors of u^*, and nodes and edges that are not shown are denoted with "...".

can be used by a greedy approach to produce approximately optimal solutions. In the following, we outline the greedy approach and present the PRH measure.

The greedy approach constructs the subset of edges $E' \subseteq E$ to be deleted iteratively. In each iteration, the approach adds into E' the edge e that minimizes the ratio of: (I) the distance between $PR(G'_1)$ and $PR(G'_2)$, where G'_1 (respectively, G'_2) is produced from the graph G by deleting E' (respectively, $E' \cup e$), and (II) *aggregate path probability gain*. The measure of the distance must be *monotone* (i.e., its value for the deletion of E' must not be larger than that for the deletion of $E' \cup e$). Otherwise, the greedy approach does not offer approximation guarantees, as we will explain later. However, the measures that capture changes to the PageRank distribution are *not* monotone, as shown in Example 3.

Example 3. *The subgraphs G'_1 and G'_2 of the graph G in Fig. 1(a) are produced by deleting $E' = \{e_1\}$ and $E' \cup e_2 = \{e_1, e_2\}$, respectively. The distance between $PR(G)$ and $PR(G'_1)$ is higher than that between $PR(G)$ and $PR(G'_2)$, according to each of the measures in Fig. 3.*

Subgraph	Deleted edges	L_1	*Gower*	*ARE*	KL-divergence
G'_1	$E' = \{e_1\}$	0.347	0.032	0.341	0.126
G'_2	$E' \cup e_2 = \{e_1, e_2\}$	0.188	0.017	0.178	0.051

Fig. 3: Existing measures favor the deletion of the edges $\{e_1, e_2\}$ instead of $\{e_1\}$ from G in Fig. 1(a).

On the contrary, PRH is a monotone measure. In Section III-A, we define the PRH of an edge $e = (u, u')$ and show that it is an effective proxy of the changes to the PageRank scores of nodes caused by deleting e.

In Section III-B, we define the PRH of a subset of edges, based on the observation that the dependencies among the PRH of these edges are weak. That is, the deletion of an edge $e = (u, u')$ does not substantially affect the PRH of another edge $e' = (\tilde{u}_1, \tilde{u}_2)$ in the subset. Specifically, we show that the change to the PageRank score of $\tilde{u}_1$ decreases exponentially with the length of the path from u to $\tilde{u}_1$.

A. The PRH of a single edge

The PRH of an edge $e = (u, u')$ is defined as $PRH(e) = (1-\alpha) \cdot \frac{PR(u,G)}{|n^+(u)|}$, where $PR(u, G)$ is the PageRank score of u in G, $|n^+(u)|$ is the out-degree of u, and α is the restart probability of Eq. 1.

Let $\delta(u^*) = PR(u^*, G) - PR(u^*, G')$ be the change to the PageRank score of a node u^*, when the deletion of the edge $e = (u, u')$ from G produces G'. We show that $PRH(e)$ can be used as a proxy of $\delta(u^*)$. Specifically, there are two cases when the edge e is deleted, which are illustrated in Fig. 2:

I u^* is an *out-neighbor* of u, and
(a) $u^* = u'$, or (b) $u^* \neq u'$.

II u^* is *not* an out-neighbor of u.

We now consider these cases in detail.

Case I Consider the case I(a). Before the deletion of e, the contribution of e to $PR(u^*)$ was $(1-\alpha) \cdot \frac{PR(u,G)}{|n^+(u)|} = PRH(e)$, according to Eq. 1. However, after deleting e, u is no longer an in-neighbor of u^*. Thus, the contribution of e to $PR(u^*)$ is zero. Now consider the case I(b). The deletion of e reduces the out-degree of the node u by one. Thus, the contribution of (u, u^*) to $PR(u^*)$ changes from $(1-\alpha) \cdot \frac{PR(u,G)}{|n^+(u)|}$ to $(1-\alpha) \cdot \frac{PR(u,G')}{|n^+(u)|-1}$. However, in either case, u^* may have a set of in-neighbors other than u, which is denoted with U_L (see Figs. 2I(a) and 2I(b)). Therefore, $\delta(u^*)$ is computed as in Eq. 2:

$$\delta(u^*) = \begin{cases} PRH(e) + (1-\alpha) \cdot \sum_{u_l \in U_L} \frac{\delta(u_l)}{|n^+(u_l)|} & , u^* = u' \\ PRH(e) - (1-\alpha) \cdot \frac{PR(u,G')}{|n^+(u)|-1} + (1-\alpha) \cdot \sum_{u_l \in U_L} \frac{\delta(u_l)}{|n^+(u_l)|} & , u^* \neq u' \end{cases} \quad (2)$$

where $\delta(u_l)$ is the change to the PageRank score of a node u_l in U_L, and α is the restart probability of Eq. 1. The proof of Eq. 2 follows easily from Eq. 1 and the definition of PRH, and it is omitted.

Case II The deletion of e changes the PageRank scores of the out-neighbors of u, according to Case I (see Figs. 2II(a) and II(b)), and the change is propagated to other nodes similarly. In particular, $\delta(u^*)$ is computed using Eq. 3:

$$\delta(u^*) = (1-\alpha) \cdot \textstyle\sum_{u_l \in n^-(u^*)} \frac{\delta(u_l)}{|n^+(u_l)|} \quad (3)$$

which follows from Eq. 2, when u is not an in-neighbor of u^*. Eq. 3 is computed backwards recursively to the out-neighbors of u.

Thus, in Cases I and II, $\delta(u^*)$ is determined by $PRH(e)$ and/or by the change to $PR(u^*)$, caused by the incoming edges to u^*. Furthermore, the change incurred by an edge (u_l, u^*) decreases exponentially with the length of the path from u to u_l. Specifically, given a simple path $q = [(u, u'), (u', u'_2) \ldots, (u'_{|q|-1}, u_l)]$ (see Fig. 2II(a)), we obtain Eq. 4:

$$\delta(u_l) = (1-\alpha)^{|q|-1} \cdot \frac{\delta(u')}{|n^+(u')| \cdot \prod_{r=2}^{|q|-1} |n^+(u'_r)|} \quad (4)$$

by recursively applying Eq. 3 for $\delta(u_l)$ over $u'_{|q|-1}, \ldots, u'_2$. The case of a path q containing a cycle is similar (omitted). Therefore, $\delta(u_l)$ diminishes as we move away from u, and $\delta(u^*)$ heavily depends on $PRH(e)$ in most cases. Consequently, PRH is a proxy of the change, caused by edge deletion, to the PageRank scores of nodes.

B. The PRH of a subset of edges

We define the PRH of an edge subset $E' \subseteq E$ as $PRH(E') = \sum_{e \in E'} PRH(e)$, where e is an edge in E' that starts from a node u of G and $PRH(e) = (1-\alpha) \cdot \frac{PR(u,G)}{|n^+(u)|}$. Clearly, PRH is monotone since $PRH(E') \leq PRH(E' \cup e')$, for each edge $e' \notin E'$.

The PRH of each edge in E' is computed based on the graph G. This strategy allows our approach to select an edge efficiently, without computing the PageRank distribution of the graph that is produced by the deletion of the currently selected edges. Furthermore, the strategy is effective, because the deletion of a currently selected edge $e = (u, u')$ does not substantially affect the PRH of another edge $e' = (\tilde{u}_1, \tilde{u}_2)$.

This is because $\delta(\tilde{u}_1)$ decreases exponentially with the length of the path from u to $\tilde{u}_1$, since $\delta(\tilde{u}_1)$ is computed by applying Eq. 4 for $u_l = \tilde{u}_1$. Thus, $\delta(\tilde{u}_1)$ is a small fraction of $\delta(u^*)$, which is already small, since $\delta(u^*)$ depends on $PRH(e)$ and our approach selects edges with small PRH. In Section VIII, we show that our PRH computation strategy is much more efficient and equally effective as the alternative strategy, which computes $PRH(e)$ on the graph that is produced from G by deleting the currently selected edges.

IV. Problem Definition

The *PED* problem is defined as follows.

Problem 1 (PageRank-preserving Edge Deletion (*PED*)). *Given a graph $G(V, E)$, a threshold $maxP$ in $[0, 1]$, a set of seed nodes S and a set of vulnerable nodes D, such that $S, D \subseteq V$ and $S \cap D = \varnothing$, and the PRH of each edge $e \in E$, find an edge subset $E' \subseteq E$, so that:* (I) *$PRH(E')$ is minimum, and* (II) *the activation probability $\mathcal{P}_v \leq maxP$, for each node $v \in D$, after the deletion of E' from G.*

The problem requires finding an edge subset E' with minimum PRH, whose deletion limits the activation probability $\mathcal{P}_v$ of each vulnerable node v to at most $maxP$. We assume that the operator selects the seeds (e.g., using existing methods [10], [13]), as well as the threshold $maxP$ and the vulnerable nodes, based on domain knowledge (e.g., customer vulnerability analysis and policies [19]). In addition, the operator computes the PRH of each edge. The *PED* problem is NP-hard, as shown in Theorem 1. Variations of the *PED* problem that use a fixed $maxP = 0$, or multiple thresholds, are easily dealt with by our algorithms.

Theorem 1. PED *is NP-hard.*

Proof. The proof is by reducing the NP-hard *Weighted Set Cover* (*WSC*) problem [8] to *PED*. The *WSC* problem is defined as follows. Given a collection $L = \{L_1, \ldots, L_m\}$ of sets, such that each $L_j \in L$ has a nonnegative weight $w(L_j)$ and $\cup_{L_j \in L} L_j = U = \{u_1, \ldots, u_n\}$, find a subcollection $L' \subseteq L$ that (I) covers all elements of U (i.e., $\cup_{L_j \in L'} L_j = U$), and (II) has minimum $\sum_{L_j \in L'} w(L_j)$.

We map a given instance $\mathcal{I}_{WSC}$ of *WSC* to an instance $\mathcal{I}_{PED}$ of *PED*, in polynomial time, as follows (see Fig. 4):

(I) Each subset $L_j \in L$ is mapped to $[s_j, x_j, (s_j, x_j)]$, where s_j is a seed, x_j is a non-seed node, and (s_j, x_j) is an edge.
(II) Each element u_i in each $L_j \in L$ is mapped to $[x_j, u_i, (x_j, u_i)]$, where u_i is a vulnerable node and (x_j, u_i) is an edge.
(III) We assign PRH to edges as follows: $PRH((s_j, x_j)) = w(L_j)$ and $PRH((x_j, u_i)) = \infty$, to force the algorithm for *PED* to select (s_j, x_j), which corresponds to L_j, and prevent the selection of (x_j, u_i).
(IV) We assign edge probabilities as follows: $p((s_j, x_j)) = 1$ and $p((x_j, u_i)) = \frac{1}{|n^-(u_i)|}$, to ensure that the path probability of $[(s_j, x_j), (x_j, u_i)]$ is determined by $|n^-(u_i)|$, which corresponds to the frequency of the element u_i over the subsets of L (number of subsets containing u_i).
(V) We set $maxP = 1 - \frac{1}{max_{u_i} |n^-(u_i)|}$, so that at least one path $[(s_j, x_j), (x_j, u_i)]$ to each u_i is disconnected after the deletion of the selected edges by the algorithm for *PED*. This corresponds to covering each element $u_i \in U$ by at least one subset L_j.

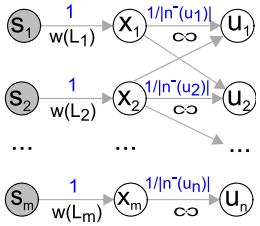


Fig. 4: The graph created from an instance of the *WSC* problem. The seeds are $s_1, \ldots, s_m$ and the vulnerable nodes are $u_1, \ldots, u_n$. The edge probability (resp. PRH) appears above (resp. below) the edges.

In the following, we prove the correspondence between a solution L' to the given instance $\mathcal{I}_{WSC}$ and a solution E' to the instance $\mathcal{I}_{PED}$.

We first prove that, if L' is a solution to $\mathcal{I}_{WSC}$, then E' is a solution to $\mathcal{I}_{PED}$. Since $\cup_{L_j \in L'} L_j = U = \{u_1, \ldots, u_n\}$, the deletion of E' disconnects at least one path to each u_i, $i \in [1, n]$, and $\mathcal{P}_{u_i} \leq maxP$ holds, for each u_i. Since $\sum_{L_j \in L'} w(L_j)$ is minimum, $PRH(E') = \sum_{L_j \in L'} w(L_j)$ is minimum. Thus, E' is a solution to $\mathcal{I}_{PED}$.

We now prove that, if an edge subset E' is a solution to $\mathcal{I}_{PED}$, then L' is a solution to $\mathcal{I}_{WSC}$. Since E' is a solution to $\mathcal{I}_{PED}$, at least one path to each u_i is disconnected, and $\mathcal{P}_{u_i} \leq maxP$ holds for each u_i, $i \in [1, n]$. Thus, L' satisfies $\cup_{L_j \in L'} L_j = \{u_1, \ldots, u_n\} = U$. Since $PRH(E') = \sum_{L_j \in L'} w(L_j)$ is minimum, L' has minimum $\sum_{L_j \in L'} w(L_j)$. Thus, L' is a solution to $\mathcal{I}_{WSC}$. □

V. Addressing *PED* for a single vulnerable node

This section details our methodology for addressing *PED*, when there is a single vulnerable node v. The main idea is to model *PED* as a *Submodular Set Cover* (*SSC*) [9], [21] problem and to develop an algorithm for *PED* based on the approximation algorithm for *SSC* [21]. Our algorithm is called GDE and is applied to the activation graph $\tilde{G}_v$ of v. The use of $\tilde{G}_v$ improves efficiency, since only edges that affect the activation probability of v are considered (see Section II-C).

Modeling *PED* as *SSC*. We show that *PED*, for a single vulnerable node, can be modeled as an *SSC* problem, by means of a reduction. We first provide the definition of the *SSC* problem [9] and then a formulation of *PED* based on *SSC*, which is referred to as *PED*$_{\text{SSC}}$ and is used in the reduction. After that, we present the reduction from *PED*$_{\text{SSC}}$ to *SSC*.

Definition 1 (Submodular Set Cover (SSC) [9])**.** *Let U be a universe of elements and $c(u)$ be the nonnegative cost of an element u of U. Let also C be a function defined as $C(\mathcal{S}) = \sum_{u\in\mathcal{S}} c(u)$, for a subset $\mathcal{S}$ of U, and F be a monotone non-decreasing submodular function. Given a nonnegative constant b, find a subset $\mathcal{S} \subseteq U$ whose cost $C(\mathcal{S})$ is minimum and $F(\mathcal{S}) \geq b$.*

The *PED*$_{\text{SSC}}$ problem is defined as follows.

Problem 2 (PED$_{\text{SSC}}$)**.** *Given a threshold $max\mathcal{P}$ in $[0, 1]$, a set of seed nodes S, a vulnerable node v, the activation graph $\tilde{G}_v$, and the PRH of each edge of $\tilde{G}_v$, find an edge subset $E' \subseteq E$ of $\tilde{G}_v$, such that:* (I) *$PRH(E')$ is minimum, and* (II) *$\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E') \leq max\mathcal{P}$, where $\mathcal{P}(v, Q_{S,v}, E)$ (resp., $\mathcal{P}(v, Q_{S,v}, E')$) is the activation probability of v by the paths of $Q_{S,v}$* that contain edges in E (*resp., in E'*).

In order to perform the reduction, we show that $\mathcal{P}(v, Q_{S,v}, E')$ is monotone non-decreasing submodular, in Theorem 2. Intuitively, the submodularity property holds, because the addition of an edge e into E' increases $\mathcal{P}(v, Q_{S,v}, E')$ by the sum of the path probabilities of paths that contain e and no other edge in E'. Thus, the increase caused by adding e into E' is at least equal to the increase to $\mathcal{P}(v, Q_{S,v}, E'')$ caused by adding e into a superset E'' of E'.

Theorem 2. *The function $\mathcal{P}(v, Q_{S,v}, E')$ is monotone non-decreasing submodular.*

Proof: Let E_v be the edge set of $\tilde{G}_v$, $E_1 \subseteq E_2 \subseteq E_v$ be subsets of E_v, and e be an edge in $E_v \setminus E_2$. Let also $Q^{E_1}_{S,v} \subseteq Q_{S,v}$ and $Q^{E_2}_{S,v} \subseteq Q_{S,v}$ be the set of paths containing edges in E_1 and E_2, respectively, and $Q^{e}_{S,v} \subseteq Q_{S,v}$ be the set of paths containing e. We will show that Eq. 5 holds in each of the following cases.

$$\mathcal{P}(v, Q_{S,v}, E_1 \cup e) - \mathcal{P}(v, Q_{S,v}, E_1) \geq \mathcal{P}(v, Q_{S,v}, E_2 \cup e) - \mathcal{P}(v, Q_{S,v}, E_2) \quad (5)$$

Case I: All paths in $Q^{e}_{S,v}$ are contained in $Q^{E_1}_{S,v}$. Thus, $\mathcal{P}(v, Q_{S,v}, E_1 \cup e) - \mathcal{P}(v, Q_{S,v}, E_1) = 0 \geq \mathcal{P}(v, Q_{S,v}, E_2 \cup e) - \mathcal{P}(v, Q_{S,v}, E_2) = 0$, since adding e does not change $Q^{E_1}_{S,v}$ and $Q^{E_2}_{S,v}$.

Case II: All paths in $Q^{e}_{S,v}$ are contained in $Q^{E_2}_{S,v}$ and at least one path is not contained in $Q^{E_1}_{S,v}$. Thus, $\mathcal{P}(v, Q_{S,v}, E_1 \cup e) - \mathcal{P}(v, Q_{S,v}, E_1) \geq \mathcal{P}(v, Q_{S,v}, E_2 \cup e) - \mathcal{P}(v, Q_{S,v}, E_2) = 0$, since adding e adds paths into $Q^{E_1}_{S,v}$ only.

Case III: At least one path in $Q^{e}_{S,v}$ is not contained in $Q^{E_2}_{S,v}$. Thus, $\mathcal{P}(v, Q_{S,v}, E_1 \cup e) - \mathcal{P}(v, Q_{S,v}, E_1) \geq \mathcal{P}(v, Q_{S,v}, E_2 \cup e) - \mathcal{P}(v, Q_{S,v}, E_2)$, since adding e adds into $Q^{E_1}_{S,v}$ all the paths that are added into $Q^{E_2}_{S,v}$ and the paths of $Q^{e}_{S,v}$ contained in $Q^{E_2}_{S,v} \setminus Q^{E_1}_{S,v}$.

Consequently, Eq. 5 holds in each case and $\mathcal{P}(v, Q_{S,v}, E')$ is submodular. In addition, $\mathcal{P}(v, Q_{S,v}, E')$ is monotone, since $\mathcal{P}(v, Q_{S,v}, E') \leq \mathcal{P}(v, Q_{S,v}, E' \cup e)$ for each $e \notin E'$, and non-decreasing, since $\mathcal{P}(v, Q_{S,v}, E_1) \leq \mathcal{P}(v, Q_{S,v}, E_2)$. □

We now present the reduction from *PED*$_{\text{SSC}}$ to *SSC*.

Theorem 3. PED$_{SSC}$ *can be reduced to* SSC.

Proof. (*Sketch*) For any instance $\mathcal{I}_{PED_{SSC}}$ of *PED*$_{\text{SSC}}$, an instance $\mathcal{I}_{SSC}$ of *SSC* can be constructed as follows: (I) for each edge e with $PRH(e)$ in the activation graph $\tilde{G}_v$, add into the universe U an element u with cost $c(u) = PRH(e)$, (II) define the function $F(\mathcal{S}) = \mathcal{P}(v, Q_{S,v}, E')$, where E' is the edge subset corresponding to $\mathcal{S} \subseteq U$, and (III) set $b = \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}$. In addition, for any feasible solution $\mathcal{S}$ of $\mathcal{I}_{SSC}$, a feasible solution E' of $\mathcal{I}_{PED_{SSC}}$ with $PRH(E') = C(\mathcal{S})$ can be constructed, by adding into E' the edges that correspond to the elements of $\mathcal{S}$. Note that E' is a solution of $\mathcal{I}_{PED_{SSC}}$ because $F(\mathcal{S}) \geq b$ implies $\mathcal{P}(v, Q_{S,v}, E') \geq \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}$, which implies $\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E') \leq max\mathcal{P}$. □

Since *PED*$_{\text{SSC}}$ can be modeled as an *SSC* problem, we can obtain an approximate solution to *PED*$_{\text{SSC}}$ using the algorithm of [21]. This algorithm iteratively adds the element $u \in U \setminus \mathcal{S}$ with the minimum ratio $\frac{c(u)}{F(\mathcal{S}\cup u) - F(\mathcal{S})}$ into $\mathcal{S}$, until $F(\mathcal{S}) \geq b$.

Note that the function C in *SSC* is monotone. Thus, *PED*$_{\text{SSC}}$ cannot be reduced to *SSC*, if a non-monotone measure is used instead of PRH.

Algorithm: GDE
Input: Graph G, activation graph $\tilde{G}_v$, threshold $max\mathcal{P}$, PageRank distribution $PR(G)$, restart probability α
Output: Set of edges E'
1 **foreach** *edge* $e = (u, u')$ *of* $\tilde{G}_v$ **do**
2 $PRH(e) \leftarrow (1 - \alpha) \cdot \frac{PR(u)}{|n^+(u)|}$
3 $E' \leftarrow \varnothing$
4 **while** $\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E') > max\mathcal{P}$ **do**
5 Reconstruct $\tilde{G}_v$ and find an edge e of $\tilde{G}_v$ s.t. $g_{\mathcal{P}}(e) > 0$ and $\frac{PRH(e)}{g_{\mathcal{P}}(e)}$ is minimum
6 $E' \leftarrow E' \cup e$
7 Delete E' from G
8 **return** E'

Greedy Delete Edges (GDE). GDE is applied to the activation graph $\tilde{G}_v$ of v and constructs the subset of edges E' to be deleted iteratively. As can be seen in the pseudocode, the algorithm computes the PRH of each edge in $\tilde{G}_v$ (steps 1-2) and constructs E', based on a similar criterion to that of the algorithm of [21] (steps 4-6). That is, it selects the edge e with the minimum ratio $\frac{PRH(e)}{g_{\mathcal{P}}(e)}$, where $g_{\mathcal{P}}(e) = \mathcal{P}(v, Q_{S,v}, E' \cup e) - \mathcal{P}(v, Q_{S,v}, E')$ is the *path probability gain* of e. The path probability gain $g_{\mathcal{P}}(e)$ quantifies the increase in $\mathcal{P}(v, Q_{S,v}, E')$, caused by the selection of e. Thus, the selected edge has small PRH and contributes significantly to lowering the activation probability $\mathcal{P}_v$. To ensure that $g_{\mathcal{P}}(e)$ is positive, we reconstruct $\tilde{G}_v$ in Step 5. Next, e is added into E' (step 6), and the process is repeated if the activation probability $\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E')$ exceeds $max\mathcal{P}$. Last, E' is deleted from G and returned (steps 7-8).

Theorem 4 shows that GDE finds a solution with PRH at most $1 + ln(\lambda)$ times larger than that of the optimal solution,

where λ depends on the PRH and path probability gain of the selected edges. The proof easily follows from [21] (omitted).

Theorem 4. *Let E' be the output of* GDE *and E'_{OPT} be the optimal solution to* PED_{SSC}. *It holds that $PRH(E') \leq (1 + ln(\lambda)) \cdot PRH(E'_{OPT})$, where λ is the minimum of:* (I) *the maximum ratio $\frac{g_{\mathcal{P}}(e_1)}{g_{\mathcal{P}}(e)}$,* (II) $\frac{PRH(e_\ell)}{g_{\mathcal{P}}(e_\ell)} / \frac{PRH(e_1)}{g_{\mathcal{P}}(e_1)}$, *and* (III) $\frac{\mathcal{P}(v,Q_{S,v},E')}{g_{\mathcal{P}}(e_\ell)}$, *where e_1 (resp., e_ℓ) is the edge that was first* (*resp., last*) *added into E', and e is an edge in $E' \setminus e_1$.*

GDE needs $O(|E_v| \cdot |E'| \cdot T)$ time, where E_v is the edge set of $\tilde{G}_v$, $E' \subseteq E_v$ is the set of deleted edges, and T is the maximum time needed to compute $g_{\mathcal{P}}(e)$. Specifically, step 4 is executed $O(|E'|)$ times, and step 5 needs $O(|E_v| \cdot T)$ time. In practice, $T << |E_v|$ because the activation probabilities are computed using small subgraphs of $\tilde{G}_v$ (see Section II-C).

VI. Algorithms for multiple vulnerable nodes

This section presents AGDE and IGDE, which address the *PED* problem when there are multiple vulnerable nodes.

Aggregate Greedy Delete Edges (AGDE). AGDE is an approximation algorithm, which reduces the activation probabilities of multiple vulnerable nodes simultaneously, using a single (aggregate) constraint function. This function allows us to model *PED* as an *SSC* problem and to base AGDE on the algorithm of [21]. In the following, we present the aggregate constraint function.

We aim to check whether the condition (II) of *PED* is satisfied, using a single function. This condition is written as $\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E') \leq max\mathcal{P}$, which implies $\mathcal{P}(v, Q_{S,v}, E') \geq \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}$, for each vulnerable node v. Now, we replace $\mathcal{P}(v, Q_{S,v}, E')$ by $min(\mathcal{P}(v, Q_{S,v}, E'), \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P})$. Clearly, this reduces (truncates) $\mathcal{P}(v, Q_{S,v}, E')$ to the constant $\mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}$, if only if $\mathcal{P}(v, Q_{S,v}, E') \geq \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}$, for a vulnerable node v. Thus, the condition (II) of *PED* is satisfied, if and only if $\sum_{v \in D} min(\mathcal{P}(v, Q_{S,v}, E'), \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P}) = \sum_{v \in D}(\mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P})$. Based on this observation, we define the aggregate constraint function, called *aggregate path probability*, as:

$$\mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E') = \sum_{v \in D} min(\mathcal{P}(v, Q_{S,v}, E'), \mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P})$$

where $D \subseteq V$ is the subset of vulnerable nodes.

We now present a formulation of the *PED* problem, which uses the aggregate path probability. For clarity, the problem in this formulation is referred to as $\textit{PED}_{Aggr}$.

Problem 3 (PED_{Aggr})**.** *Let $S \subseteq V$ be the subset of seed nodes, $D \subseteq V$ be the subset of vulnerable nodes, and $\tilde{G}_D = \cup_{v \in D} \tilde{G}_v$ be the activation graph of D. Given a threshold $max\mathcal{P}$ in $[0, 1]$ and the PRH of each edge of $\tilde{G}_D$, find an edge subset $E' \subseteq E$ of $\tilde{G}_D$ such that:* (I) *$PRH(E')$ is minimum, and* (II) $\mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E') = \sum_{v \in D} (\mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P})$.

$\textit{PED}_{Aggr}$ can be reduced to *SSC*, based on a similar reduction to that of Theorem 3 (omitted), where the submodularity of the function $\mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E')$ easily follows from the submodularity of $\mathcal{P}(v, Q_{S,v}, E')$. Thus, we can obtain an approximate solution to $\textit{PED}_{Aggr}$ by using the algorithm of [21] as the basis of our AGDE algorithm.

In what follows, we present the AGDE algorithm. As can be seen in the pseudocode, the algorithm is applied to the activation graph $\tilde{G}_D$ and iteratively selects the edge with the minimum ratio $\frac{PRH(e)}{g_{\mathcal{P}_D}(e)}$ (step 5), where $g_{\mathcal{P}_D}(e) = \mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E' \cup e) - \mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E')$ is the *aggregate path probability gain*. The process is repeated until the condition (II) of $\textit{PED}_{Aggr}$ holds. Note that this condition holds, in the worst case when E' contains all edges of $\tilde{G}_D$. Thus, AGDE will always terminate.

Algorithm: AGDE
Input: Graph G, activation graph $\tilde{G}_D$, threshold $max\mathcal{P}$, set of vulnerable nodes D, PageRank distribution $PR(G)$, restart probability α
Output: Set of edges E'
1 **foreach** *edge* $e = (u, u')$ *of* $\tilde{G}_D$ **do**
2 $PRH(e) \leftarrow (1 - \alpha) \cdot \frac{PR(u,G)}{|n^+(u)|}$
3 $E' \leftarrow \varnothing$
4 **while** $\mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E') < \sum_{v \in D}(\mathcal{P}(v, Q_{S,v}, E) - max\mathcal{P})$ **do**
5 Reconstruct $\tilde{G}_D$ and find an edge e of $\tilde{G}_D$ s.t. $g_{\mathcal{P}_D}(e) > 0$ and $\frac{PRH(e)}{g_{\mathcal{P}_D}(e)}$ is minimum
6 $E' \leftarrow E' \cup e$
7 Delete E' from G
8 **return** E'

AGDE finds a solution with PRH at most $1 + ln(\lambda_D)$ times larger than that of the optimal solution to $\textit{PED}_{Aggr}$, where λ_D is as in Theorem 4 but with $g_{\mathcal{P}_D}$ (respectively, $\mathcal{P}(D, \cup_{v \in D} Q_{S,v}, E')$) instead of $g_{\mathcal{P}}$ (respectively, $\mathcal{P}(v, Q_{S,v}, E')$). The proof follows from [21] and is omitted. Clearly, AGDE needs $O(|E| \cdot |E'| \cdot T_D)$ time, where E is the edge set of $\tilde{G}_D$, $E' \subseteq E$ is the set of deleted edges, and T_D is the maximum time needed to compute $g_{\mathcal{P}_D}$.

Iterative Greedy Delete Edges (IGDE). As can be seen in the pseudocode, IGDE sorts the vulnerable nodes, in decreasing order of activation probability, and applies GDE to the activation graph $\tilde{G}_v$ of one vulnerable node v at a time.

This heuristic improves efficiency, because: (I) $\tilde{G}_v$ contains a much smaller number of edges than the activation graph of all vulnerable nodes to which AGDE is applied, and (II) the edge subset E'_v that is deleted in an iteration is not considered again. However, this reduces the number of explored solutions. Therefore, vulnerable nodes with large activation probability $\mathcal{P}(v, Q_{S,v}, E)$ are dealt with first, when more edges are available for deletion.

Algorithm: IGDE
Input: Graph G, threshold $max\mathcal{P}$, set of vulnerable nodes D, activation graph $\tilde{G}_v$ for each $v \in D$, PageRank distribution $PR(G)$, restart probability α
Output: Set of edges E'
1 sort each v in D in decreasing order of activation probability $\mathcal{P}(v, Q_{S,v}, E)$
2 $E' \leftarrow \varnothing$; $G_{tmp} \leftarrow G$
3 **foreach** v *in* D **do**
4 **if** $\mathcal{P}(v, Q_{S,v}, E) - \mathcal{P}(v, Q_{S,v}, E') > max\mathcal{P}$ **then**
5 $E'_v \leftarrow$ GDE(G, $\tilde{G}_v, max\mathcal{P}, PR(G_{tmp}), \alpha$)
6 **foreach** v *in* D **do**
7 Update $\tilde{G}_v$ to reflect the deletion of E'_v
8 $E' \leftarrow E' \cup E'_v$
9 **return** E'

Note that each vulnerable node v is considered once, because $\mathcal{P}(v, Q_{S,v}, E')$ cannot decrease in the next iterations (see Theorem 2). Thus, after the loop of step 3 terminates,

the condition (II) of *PED* holds, and the subset of edges E' is returned (step 9). Furthermore, GDE is applied to the PageRank distribution of the original graph (step 5), so that edge deletion does not affect the PRH computation in GDE.

IGDE needs $O(\sum_{v \in D}(|E_v| \cdot |E'| \cdot T + |D| \cdot |E_v|))$ time.

VII. Related work

Existing methods limit the diffusion of information by modifying the graph, or by initiating the diffusion of information of opposite content.

Methods that modify the graph aim to minimize the spread (expected number of activated nodes) directly, or indirectly by optimizing a graph property. To minimize the spread directly, there are heuristics that apply node [23] or edge [15] deletion, under the Independent Cascade (IC) model (e.g., [23]), or under the Linear Threshold (LT) model (e.g., [15]). There is also an approximation algorithm [14] under the LT model, which deletes an edge subset of given size. Methods for minimizing the spread indirectly were proposed in [18], [20]. All methods that modify the graph follow the *collective* approach, which requires reducing the activation probabilities of *all* nodes as much as possible. In addition, they assume that deleting each edge has the same impact on the information propagation properties of the graph. Thus, these methods are not applicable to our problem, as discussed in Introduction.

Methods that minimize the spread of undesirable (negative) information, by diffusing information of opposite content (positive information) were proposed in [5], [12], under an extended IC [5] or LT [12] model. These methods select seeds which diffuse the positive information and aim to prevent the diffusion of negative information to the largest (expected) number of nodes. However, the *PED* problem requires limiting the diffusion to vulnerable nodes, while not affecting the information propagation to other nodes. Therefore, these methods cannot be applied to our problem.

VIII. Experimental evaluation

In this section, we evaluate AGDE and IGDE, in terms of their effectiveness and efficiency. Since existing methods are not applicable to the *PED* problem, we compared our algorithms against baselines that use different edge selection criteria, and against the optimal method, BruteForce, which examines all edge subsets. In addition, we show that PRH is an effective and efficient proxy of the change to the PageRank scores, caused by edge deletion.

Setup and datasets. To quantify the impact of edge deletion, we used: (I) PRH, (II) the L_1 distance, (III) the percentage of deleted edges, and (IV) the Kendall τ_b correlation ($K\tau_b$). $K\tau_b$ captures changes to the ranking of all nodes, with respect to their PageRank scores [3]. A $K\tau_b$ value of 1 implies no change to the ranking and larger values are preferred.

We implemented all algorithms in C++ and applied them to the *cit-HepPh* (*Ph*), *Wiki-vote* (*Wiki*), and *Polblogs* (*Pol*) datasets. *Ph* and *Wiki* are available at http://snap.stanford.edu/data and *Pol* at http://www-personal.umich.edu/~mejn/. We also used two synthetic datasets, *AB* and *ER*, which were generated by the Albert-Barabasi and the Erdös-Rényi model, respectively. These models were also used in [7], [14]. Table I summarizes the characteristics of each dataset and its default values for $max\mathcal{P}$, $|S|$ (# of seeds), and $|D|$ (# of vulnerable nodes). BruteForce does not scale to real datasets. Thus, it was applied to 1000 datasets, which have 16 nodes and 28 edges on average and were generated by the Erdös-Rényi model.

Dataset	$\|V\|$	$\|E\|$	(avg, max) in-deg.	$max\mathcal{P}$	$\|S\|$	$\|D\|$
Ph	34546	421578	(24.3, 846)	0.1	200	50
Wiki	7115	103689	(13.7, 452)	0.1	75	20
Pol	1490	19090	(11.9, 305)	0.1	500	20
AB	111150	500000	(9, 99907)	0.01	500	5
ER	5000	49917	(9.98, 24)	0.01	50	20

TABLE I: Characteristics of datasets and default values

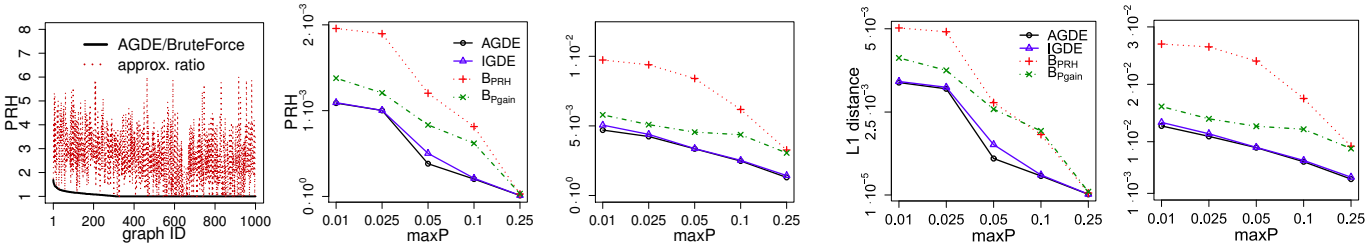
All edge probabilities were assigned by the *uniform* method (i.e., each incoming edge to u has edge probability $\frac{1}{|n^-(u)|}$) [7], [13] and the threshold h was set to 10^{-3}, as in [10]. The vulnerable nodes were: (I) selected randomly among the top-10% of nodes with the largest in-degree, in *Ph*, *Wiki*, *Pol*, and *ER*, and (II) the 5 nodes with the largest in-degree, in all other datasets. This excludes nodes that are easy to deal with. To find the seeds, we considered each vulnerable node v and iteratively selected random paths of length at least 2 that end at v, until $\mathcal{P}_v \geq min(r \cdot max\mathcal{P}, 1)$, where $r \geq 1$ is a random integer. The start nodes of these paths were used as seeds. Since there were many other paths from seeds to vulnerable nodes, the activation graphs were large. All experiments ran on an Intel Xeon at 2.4Ghz with 12Gb RAM.

Quality of approximation. We demonstrate that AGDE finds near-optimal solutions, by comparing it to BruteForce. Fig. 5a shows the ratio between the PRH for AGDE and for BruteForce, as well as the approximation ratio $1+ln(\lambda_D)$, when $max\mathcal{P} = 0.2$, for all 1000 datasets (sorted in decreasing PRH). The ratio is 1 for 70% of the datasets, 1.04 on average, and at most 1.7. The approximation ratio is 2.6 on average and at most 6. Thus, AGDE produced solutions that are close to optimal, and the ratio of AGDE to BruteForce was much lower than the approximation ratio.

Effectiveness. We demonstrate that AGDE and IGDE do not substantially affect the information propagation properties of the graph and that they delete a small number of edges. We compared our methods with two baselines: (I) B_{PRH}, which selects the edge with the minimum PRH, and (II) B_{PGain}, which selects the edge with the maximum aggregate path probability gain. Both baselines are based on AGDE but do not offer approximation guarantees.

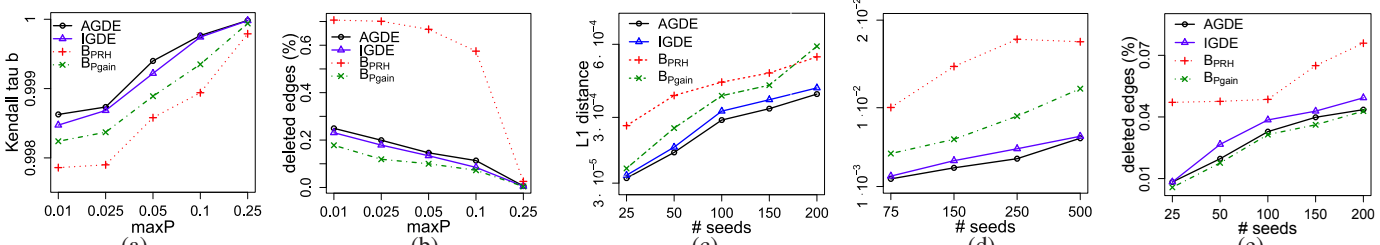
Figs. 5b and 5c show the PRH, for varying $max\mathcal{P}$. The PRH decreases as $max\mathcal{P}$ increases, because the required reduction in activation probabilities becomes smaller. AGDE was the best method, and the PRH for IGDE was slightly larger. The baselines performed much worse, because B_{PRH} deleted edges that did not reduce the activation probabilities of vulnerable nodes and B_{PGain} deleted edges with large PRH.

Figs. 5d and 5e show the results for the L_1 distance, which follows the same trend as PRH. This suggests that minimizing PRH helps preserving the PageRank distribution. AGDE and IGDE performed similarly with respect to $K\tau_b$ and better than



(a) (b) (c) (d) (e)

Fig. 5: (a) AGDE/BRUTEFORCE ratio with respect to PRH and approximation ratio of AGDE. PRH vs. $max\mathcal{P}$ for: (b) *Wiki*, (c) *Pol*. L_1 distance vs. $max\mathcal{P}$ for: (d) *Wiki*, (e) *Pol*.



(a) (b) (c) (d) (e)

Fig. 6: (a) K_{τ_b} vs. $max\mathcal{P}$, and (b) deleted edges (%) vs. $max\mathcal{P}$, for *Wiki*. L_1 distance vs. $|S|$, for: (c) *Ph*, and (d) *Pol*. (e) Percentage of deleted edges vs. $|S|$, for *Ph*.

the baselines (see Fig. 6a). Furthermore, AGDE and IGDE deleted at most 0.04% more edges than B_{PGain}, which aims to minimize the number of deleted edges (see Fig. 6b).

Next, we measured effectiveness, for varying $|S|$, using seed sets of increasing size, whose elements were contained in all larger sets. Figs. 6c and 6d show that the L_1 distance increases with $|S|$. This is because the activation probabilities of vulnerable nodes, before edge deletion, are higher for large seed sets. They also show that AGDE and IGDE outperformed both baselines. Furthermore, AGDE and IGDE deleted at most 0.01% more edges than B_{PGain} (see Fig. 6e).

We also measured effectiveness, for varying $|D|$ (# of vulnerable nodes). AGDE and IGDE performed similarly and significantly better than both baselines, with respect to the L_1 distance (see Fig. 7a). Furthermore, our methods deleted at most 0.5% more edges than B_{PGain} (see Fig. 7b).

Thus, AGDE and IGDE preserved the information propagation properties of the graph much better than both baselines, and they deleted a similar number of edges with B_{PGain}, which aims to minimize the number of deleted edges.

Efficiency. We demonstrate that AGDE and IGDE scale well with $|S|$, $|D|$, and $|E|$, and that they are more efficient than the fastest baselines, B_{PRH} and B_{PGain}. In addition, we show that IGDE is substantially more efficient than AGDE.

Fig. 7c shows that AGDE and IGDE scaled better than linear (*sublinearly*) with $|S|$. However, IGDE was up to 4 times faster, as it considers seeds contained in the activation graph of one vulnerable node at a time. Fig. 7d shows that IGDE scaled sublinearly with $|D|$, and it was up to *two orders of magnitude faster* than AGDE. This is because the edges deleted in an iteration of IGDE affected many activation graphs. Fig. 7e shows that AGDE and IGDE scaled sublinearly with $|E|$, and that IGDE was up to one order of magnitude faster. The baselines scaled similarly to AGDE, and the results for the *ER* dataset were similar (omitted).

Threshold h. We demonstrate the impact of h on the L_1 distance and on the runtime of AGDE and IGDE. Figs. 8a and 8b show that the L_1 distance decreased by 0.07% on average, for $h \leq 10^{-3}$ and substantially for larger h values. The runtime of both methods decreased significantly as h increases. Thus, setting h to 10^{-3}, as suggested in [10], allows estimating the activation probabilities accurately and efficiently.

Benefit of using PRH vs. the L_1 distance. We demonstrate the effectiveness and efficiency of using PRH as a proxy of the change to the PageRank scores, caused by edge deletion. We compared our algorithms against $\text{B}_{L1/PGain}$, a baseline that implements the greedy approach based on the L_1 distance (see Section III).

Figs. 8c, 8d, and 8e show the results for varying $max\mathcal{P}$, $|S|$, and $|D|$, respectively, with respect to the L_1 distance. The L_1 distance for $\text{B}_{L1/PGain}$ was 5 times larger than that of AGDE and IGDE, on average. In addition, $\text{B}_{L1/PGain}$ was several orders of magnitude slower than our algorithms, because it computes the PageRank distribution of the graph after deleting each edge, in order to select the best edge in each iteration. For example, $\text{B}_{L1/PGain}$ required 12 hours when applied to *Pol* with $|D| = 50$, while IGDE needed 90 seconds. Thus, PRH is a more effective and efficient measure to avoid changes to the PageRank distribution compared to the L_1 distance.

Benefit of computing PRH on G. We demonstrate that computing the PRH of every edge in a subset E' on the graph G helps efficiency and does not impact effectiveness. We compared AGDE with $\text{B}_{PRHupd/PGain}$, a baseline that computes the PRH of an edge e on a graph G'_e, produced from G by deleting all edges that were added into E' before e. The baseline is based on AGDE, because AGDE computes the PRH of more edge subsets (potential solutions) than IGDE, and this allows comparing the PRH computation strategies on more subsets. We repeated all experiments of the effectiveness subsection above and found that AGDE and

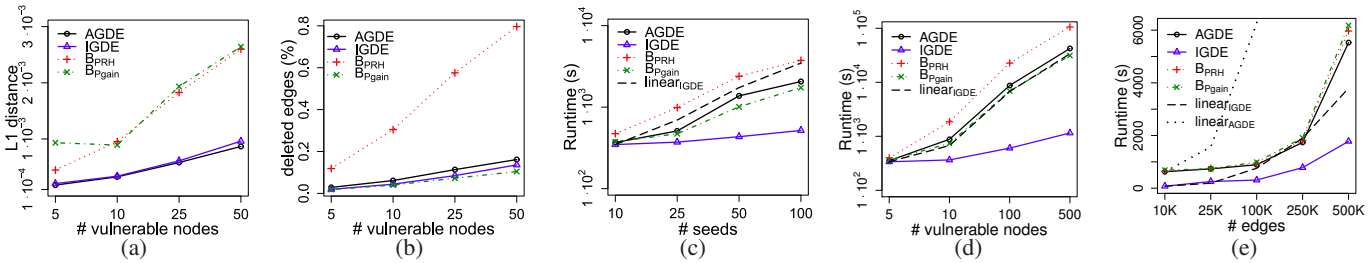


Fig. 7: (a) L_1 distance vs. $|D|$, for *Wiki*. (b) Percentage of deleted edges vs. $|D|$, for *Wiki*. Runtime vs. (c) $|S|$, and (d) $|D|$, for *Wiki*. (e) Runtime vs. $|E|$, for *AB*.

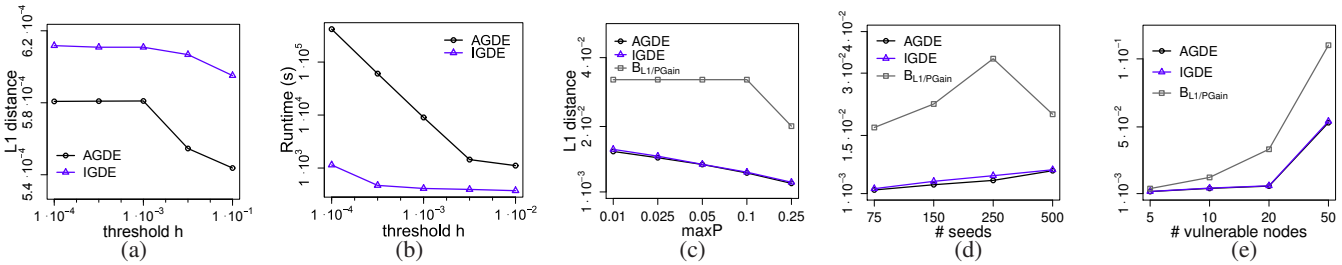


Fig. 8: (a) L_1 distance vs. h, for *Wiki*. (b) Runtime vs. h, for *Wiki*. Comparison with $\mathrm{B}_{L1/PGain}$. L_1 distance vs. (c) $max\mathcal{P}$, (d) $|S|$, and (e) $|D|$, for *Pol*.

$\mathrm{B}_{PRHupd/PGain}$ produced the same solutions, except in the experiments of Figs. 5c and 6d. In the experiments of Figs. 5c and 6d the algorithms broke ties differently. Thus, their solutions differed in at most 5 edges and had the same PRH. However, AGDE was up to 84% faster, because it avoids recomputing the PageRank distribution of the graph. Thus, our PRH computation strategy is both effective and efficient.

IX. Conclusions

Existing works for limiting the diffusion of information by edge deletion assume that the diffusing information can affect all nodes and that deleting each edge has the same impact on the information propagation properties of the graph. In this work, we introduced an approach that lifts these restrictive assumptions. Our approach reduces the activation probabilities of vulnerable nodes to at most a specified threshold, and it applies edge deletion while preserving the information propagation properties of the graph, by avoiding changes to its PageRank distribution. We proposed a measure to capture these changes, and based on the measure we formulated the *PED* problem. To deal with the problem, we developed an effective approximation algorithm and an efficient heuristic.

Acknowledgments

G. Loukides was supported by a RAENG Research Fellowship.

References

[1] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte carlo methods in pagerank computation. *SIAM J. Numer. Anal.*, 45(2):890–904, 2007.

[2] P. Berkhin. A survey on pagerank computing. *Internet Math.*, 2(1):73–120, 2005.

[3] P. Boldi, M. Santini, and S. Vigna. Paradoxical effects in pagerank incremental computations. *Internet Math.*, 2(3):387–404, 2005.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Com. Net. & ISDN*, 30(1):107–117, 1998.

[5] C. Budak, D. Agrawal, and A. E. Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, 2011.

[6] W. Chen et al. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*, 2011.

[7] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, 2010.

[8] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[9] T. Fujito. Approximation algorithms for submodular set cover with applications. *IEICE Trans. Inf. and Syst.*, e83(3), 2000.

[10] A. Goyal, L. Wei, and L. Lakshmanan. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, 2011.

[11] S. Gupta. A conceptual framework that identifies antecedents and consequences of building socially responsible international brands. *Thunderbird International Business Review*, 2015.

[12] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, 2012.

[13] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[14] E. B. Khalil, B. Dilkina, and L. Song. Scalable diffusion-aware optimization of network topology. In *KDD*, 2014.

[15] M. Kimura, K. Saito, and H. Motoda. Solving the contamination minimization problem on networks for the linear threshold model. In *PRICAI*. 2008.

[16] A. Krause and D. Golovin. Submodular function maximization. In *Tractability*. 2013.

[17] A. N. Langville and C. D. Meyer. A survey of eigenvector methods for web information retrieval. *SIAM Review*, 47(1):135–161, 2005.

[18] S. Saha, A. Adiga, B. A. Prakash, and A. K. S. Vullikanti. Approximation algorithms for reducing the spectral radius to control epidemic spread. In *SDM*, 2015.

[19] N. C. Smith and E. Cooper-Martin. Ethics and target marketing: The role of product harm and consumer vulnerability. *Journal of Marketing*, 61(3):pp. 1–20, 1997.

[20] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM*, 2012.

[21] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[22] B. Xiang et al. Pagerank with priors: An influence propagation perspective. In *IJCAI*, pages 2740–2746, 2013.

[23] Y. Zhang and B. A. Prakash. DAVA: distributing vaccines over networks under prior information. In *SDM*, 2014.