Reverse Prevention Sampling for Misinformation Mitigation in Social Networks

Michael Simpson
Department of Computer Science, University of Victoria, Canada
simpsonm@uvic.ca

Venkatesh Srinivasan
Department of Computer Science, University of Victoria, Canada
srinivas@uvic.ca

Alex Thomo
Department of Computer Science, University of Victoria, Canada
thomo@uvic.ca

Abstract

In this work, we consider misinformation propagating through a social network and study the problem of its prevention. In this problem, a "bad" campaign starts propagating from a set of seed nodes in the network and we use the notion of a limiting (or "good") campaign to counteract the effect of misinformation. The goal is to identify a set of k users that need to be convinced to adopt the limiting campaign so as to minimize the number of people that adopt the "bad" campaign at the end of both propagation processes.

This work presents *RPS* (Reverse Prevention Sampling), an algorithm that provides a scalable solution to the misinformation prevention problem. Our theoretical analysis shows that *RPS* runs in $O((k+l)(n+m)(\frac{1}{1-\gamma})\log n/\epsilon^2)$ expected time and returns a $(1-1/e-\epsilon)$-approximate solution with at least $1-n^{-l}$ probability (where γ is a typically small network parameter and l is a confidence parameter). The time complexity of *RPS* substantially improves upon the previously best-known algorithms that run in time $\Omega(mnk \cdot POLY(\epsilon^{-1}))$. We experimentally evaluate *RPS* on large datasets and show that it outperforms the state-of-the-art solution by several orders of magnitude in terms of running time. This demonstrates that misinformation prevention can be made practical while still offering strong theoretical guarantees.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Approximation algorithms analysis

Keywords and phrases Graph Algorithms, Social Networks, Misinformation Prevention

Digital Object Identifier 10.4230/LIPIcs.ICDT.2020.24

Related Version A full version of the paper is available at `https://arxiv.org/abs/1807.01162`.

1 Introduction

Social networks allow for widespread distribution of knowledge and information in modern society as they have rapidly become a place to hear the news and discuss social topics. Information can spread quickly through the network, eventually reaching a large audience, especially so for influential users. While the ease of information propagation in social networks can be beneficial, it can also have disruptive effects. In recent years, the number of high profile instances of misinformation causing severe real-world effects has risen sharply. These examples range across a number of social media platforms and topics [9, 23, 11, 13, 29, 1]. Thus, in order for social networks to serve as a reliable platform for disseminating critical information, it is necessary to have tools to limit the spread of misinformation.

Budak et al. [4] were among the first to formulate the problem of misinformation prevention as a combinatorial optimization problem. By building upon the seminal work of Kempe et al. [16] on *influence maximization* to a model that can handle multiple campaigns ("bad" and "good"), they present a greedy approach that provides a $(1 - 1/e - \epsilon)$-approximate solution. Unfortunately, the greedy approach of [4] is plagued by the same scaling issues as [16] when considering large social networks and is further exacerbated by the added complexity of tracking multiple cascades which requires costly shortest path computations. This leads us to the motivating question for this paper: Can we find scalable algorithms for the misinformation prevention problem introduced in [4]?

The scalability hurdle in the single campaign setting was recently resolved by Borgs et al. [3] when they made a theoretical breakthrough that fundamentally shifts the way in which we view the influence maximization problem. Their key insight was to reverse the question of "what subset of the network can a particular user influence" to "who could have influenced a particular user". Their sampling method runs in close to linear time and returns a $(1 - 1/e - \epsilon)$-approximate solution with at least $1 - n^{-l}$ probability. In addition, Tang et al. [30] presented a significant advance that improved the practical efficiency of Borgs et al. through a careful theoretical analysis that rids their approach of a large hidden constant in the runtime guarantee. Borgs et al. [3] leave open the question whether their framework can be extended to other influence propagation models.

In this work, we resolve the question of [3] for the misinformation prevention problem and achieve scalability in the multi-campaign model. We complement our theoretical analysis with extensive experiments which show an improvement of several orders of magnitude over Budak et al. [4]. Since influence in the single campaign setting corresponds to reachability in the network, our solution requires mapping the concept of reachability to an analogous notion in the multi-campaign model for misinformation prevention. Our first contribution is to show that reachability alone is not sufficient in determining the ability to save a particular node from the bad campaign. In order to address this challenge, we introduce a crucial notion of "obstructed" nodes, which are nodes such that all paths leading to them can be blocked by the bad campaign.

Using our newly defined notion of obstruction, we develop an efficient algorithm for the misinformation prevention problem that provides much improved scalability over the existing Monte Carlo-based greedy approach of [4]. A novel component of this algorithm is a procedure to compute the set of unobstructed nodes that could have saved a particular node from adopting the misinformation. We obtain theoretical guarantees on the expected runtime and solution quality for our new approach and show that its expected runtime substantially improves upon the expected runtime of [4]. Additionally, we rule out sublinear algorithms for our problem through a lower bound on the time required to obtain a constant approximation.

Finally, from an experimental point of view, we show that our algorithm gives a significant improvement over the state of the art algorithm and can efficiently handle graphs with more than 50 million edges. In summary, the contributions of this paper are:

1. We introduce the concept of *obstructed* nodes that fully captures the necessary conditions for preventing the adoption of misinformation in the multi-campaign model. In the process, we close a gap in the work of [4].
2. We design and implement a novel procedure for computing the set of nodes that could save a particular user from adopting the misinformation.
3. We propose a misinformation prevention approach that returns a $(1-1/e-\epsilon)$-approximate solution with high probability in the multi-campaign model and show that its expected runtime substantially improves upon that of the algorithm of Budak et al. [4].

4. We give a lower bound of $\Omega(m+n)$ on the time required to obtain a constant approximation for the misinformation prevention problem.
5. Our experiments show that our algorithm gives an improvement of several orders of magnitude over Budak et al. [4] and can handle graphs with more than 50 million edges.

2 Related Work

There exists a large body of work on the Influence Maximization problem first proposed by Kempe et al. [16]. The primary focus of the research community has been related to improving the practical efficiency of the Monte Carlo-based greedy approach under the Independent Cascade (IC) or Linear Threshold (LT) propagation models. These works fall into two categories: heuristics that trade efficiency for approximation guarantees [15, 32] and practical optimizations that speed up the Monte Carlo-based greedy approach while retaining the approximation guarantees [18, 6, 10]. Despite these advancements, it remains infeasible to scale the Monte Carlo-based approach to web-scale networks.

Borgs' et al. [3] brought the first asymptotic runtime improvements while maintaining the $(1 - 1/e - \epsilon)$-approximation guarantees with their *reverse influence sampling* technique. Furthermore, they prove their approach is near-optimal under the IC model. Tang et al. [30] presented practical and theoretical improvements to the approach and introduced novel heuristics that result in up to 100-fold improvements to the runtime.

Incorporating the spread of multiple campaigns is split between two main lines of work: (1) studying influence maximization in the presence of competing campaigns [2, 20, 24, 19] and (2) limiting the spread of misinformation and rumours by launching a truth campaign [4, 14, 7, 22]. In both cases, existing propagation models (such as IC and LT) are augmented or extended. The work of [4] best captures the idea of preventing the spread of misinformation in a multi-campaign version of the IC model since they aim to minimize the number of users that end up adopting the misinformation. Unfortunately, despite the objective function proving to be monotone and submodular, the Monte Carlo-based greedy solution used in [4] faces the same challenges surrounding scalability as [16].

Works [20, 8] extend the *reverse influence sampling* technique of [3] to competing campaigns (such as two competing products in [20] and spreading truth to combat misinformation in [8]). However, their work differs from ours in an important way: they use a model, different from ours, where the edge probabilities are *campaign oblivious*. This alternative model does not capture the notion of misinformation as well as the model we use, but instead is better suited for the influence maximization problem when there are multiple competing campaigns (see [4] for a discussion).

Finally, the misinformation problem has been tackled by a wide range of communities such as [17, 26, 25, 12, 31, 27].

3 Preliminaries

In this section, we formally define the multi-campaign diffusion model, the eventual influence limitation problem presented by Budak et al. [4], and present an overview of the state-of-the-art reverse sampling approach [16, 3, 30] for the influence maximization problem.

Diffusion Model

Let C (for "bad Campaign") and L (for "Limiting") denote two influence campaigns. Let $\mathcal{G} = (V, E, p)$ be a social network with node set V and directed edge set E ($|V| = n$ and $|E| = m$) where p specifies campaign-specific pairwise influence probabilities (or weights)

between nodes. That is, $p : E \times Z \to [0, 1]$ where $Z \in \{C, L\}$. For convenience, we use $p_Z(e)$ for $p(e, Z)$. Further, let $G = (V, E)$ denote the underlying unweighted directed graph. Given $\mathcal{G}$, the Multi-Campaign Independent Cascade model (MCIC) of Budak et al. [4] considers a time-stamped influence propagation process as follows:

1. At timestamp 1, we *activate* selected sets A_C and A_L of nodes in $\mathcal{G}$ for campaigns C and L respectively, while setting all other nodes *inactive*.
2. If a node u is first activated at timestamp i in campaign C (or L), then for each directed edge e that points from u to an inactive neighbour v in C (or L), u has $p_C(e)$ (or $p_L(e)$) probability to activate v at timestamp $i + 1$. After timestamp $i + 1$, u cannot activate any node.
3. In the case when two or more nodes from different campaigns are trying to activate v at a given time step we assume that the "good information" (i.e. campaign L) takes effect.
4. Once a node becomes activated in one campaign, it never becomes inactive or changes campaigns.

He et al. [14] consider the opposite policy to (3) where the misinformation succeeds in the case of a tie-break. We note that our algorithms presented in this work are applicable for both choices of the tie-break policy.

3.1 Formal Problem Statement

A natural objective, as outlined in [4], is "saving" as many nodes as possible. That is, we seek to minimize the number of nodes that end up adopting campaign C when the propagation process is complete. This is referred to as the *eventual influence limitation problem (EIL)*.

Let A_C and A_L be the set of nodes from which campaigns C and L start, respectively. Let $I(A_C)$ be the set of nodes that are activated in campaign C in the absence of L when the above propagation process converges and $\pi(A_L)$ be the size of the subset of $I(A_C)$ that campaign L prevents from adopting campaign C. We refer to A_L and A_C as the *seed sets*, $I(A_C)$ as the *influence* of campaign C, and $\pi(A_L)$ as the *prevention* of campaign L. The nodes that are prevented from adopting campaign C are referred to as *saved*. Note that $\pi(A_L)$ is a random variable that depends on the edge probabilities that each node uses in determining out-neighbors to activate.

Budak et al. [4] present a simplified version of the problem that captures the idea that it may be much easier to convince a user of the truth. Specifically, the information from campaign L is accepted by users with probability 1 ($p_L(e) = 1$ if edge e exists and $p_L(e) = 0$ otherwise) referred to as the *high effectiveness property*. In [4] it is shown that even with these restrictions EIL with the high effectiveness property is NP-hard. Interestingly, with the high effectiveness property, the prevention function is submodular and thus a Monte Carlo-based greedy approach (referred to here as *MCGreedy*) yields approximation guarantees.

We motivate the high effectiveness property with the following two real-world scenarios: (1) the phenomenon of "death hoaxes" (where celebrities or other notable figures are claimed to have died) have a strong corrective measure when the victim, or a close relative, makes an announcement on their personal account that contradicts the rumour and (2) false reporting of natural disasters can be countered by trusted news organizations providing coverage of the location of the purported scene. In both cases, the sharing of links to strong video, photographic, or text evidence that is also coming from a credible source lends itself to a scenario following the high effectiveness property. In addition to the scenarios we have outlined, the model is attractive because this assumption leads to interesting theoretical guarantees. Budak et al. study and obtain results for EIL with the high effectiveness property and is the problem that we consider in this work.

▶ **Problem 1.** *Given $\mathcal{G}$, seed set A_C, and a positive integer k, the eventual influence limitation (EIL) problem asks for a size-k seed set A_L maximizing the value of $\mathbb{E}[\pi(A_L)]$ under the MCIC model with the high effectiveness property.*

Possible Worlds Interpretation

To facilitate a better understanding of MCIC, we define a *Possible World (PW) model* that provides an equivalent view of the MCIC model and follows a widely used convention when studying IM and related problems [16, 4, 6, 10, 20, 14, 4, 7, 22]. Given a graph $\mathcal{G} = (V, E, p)$ and the MCIC diffusion model, a possible world X consists of two *deterministic graphs*, one for each campaign, sampled from a probability distribution over $\mathcal{G}$. The stochastic diffusion process under the MCIC model has the following equivalent description: we can interpret $\mathcal{G}$ as a distribution over unweighted directed graphs, where each edge e is independently realized with probability $p_C(e)$ (or $p_L(e)$). Observe, given the high effectivness property, the deterministic graph that defines the possible world for campaign L is simply the underlying unweighted graph G. Then, if we realize a graph g according to the probability distribution given by $p_C(e)$, we can associate the set of saved nodes in the original process with the set of nodes which campaign L reaches before campaign C during a *deterministic* diffusion process in $g \sim \mathcal{G}$ by campaign C and in G by campaign L. That is, we can compute the set of saved nodes with a deterministic cascade in the resulting possible world $X = (g, G)$. The following theorem from [5] establishes the equivalence between this possible world model and MCIC. This alternative PW model formulation of the EIL problem under the MCIC model will be used throughout the paper.

▶ **Theorem 1** ([5]). *For any fixed seed sets A_C and A_L, the joint distributions of the sets of C-activated nodes and L-activated nodes obtained (i) by running a MCIC diffusion from A_C and A_L and (ii) by randomly sampling a possible world $X = (g, G)$ and running a deterministic cascade from A_C in g and A_L in G, are the same.*

3.2 Reverse Sampling for Influence Maximization

In this section we review the state-of-the-art approach to the well studied *influence maximization problem (IM)*. This problem is posed in the popular Independent Cascade model (IC) which, unlike the MCIC model, only considers a single campaign. The goal here is to compute a seed set S_{IM} of size k that maximizes the influence of S_{IM} in $\mathcal{G}$. In a small abuse of notation, this section refers to a possible world as the single deterministic graph $g \sim \mathcal{G}$ where each edge in $\mathcal{G}$ is associated with a single influence probability $p(e)$.

Borgs et al. [3] were the first to propose a novel method for solving the IM problem under the IC model that avoids the limitations of the original Monte Carlo-based solution [16]. Their approach, which was later refined by Tang et al. [30], is based on the concept of *Reverse Reachable (RR) sets* and is orders of magnitude faster than the greedy algorithm with Monte Carlo simulations, while still providing approximation guarantees with high probability. We follow the convention of [30] and refer to the method of [3] as *Reverse Influence Sampling (RIS)*. To explain how *RIS* works, Tang et al. [30] introduce the following definitions:

▶ **Definition 1** (Reverse Reachable Set). *The reverse reachable set for a node v in $g \sim \mathcal{G}$ is the set of nodes that can reach v. (That is, for each node u in the RR set, there is a directed path from u to v in g.)*

▶ **Definition 2** (Random RR Set). *A random RR set is an RR set generated on an instance of $g \sim \mathcal{G}$, for a node selected uniformly at random from g.*

Note, a random RR set encapsulates two levels of randomness: (i) a deterministic graph $g \sim \mathcal{G}$ is sampled where each edge $e \in E$ is independently removed with probability $(1-p(e))$, and (ii) a "root" node v is randomly chosen from g. The connection between RR sets and node activation is formalized in the following crucial lemma.

▶ **Lemma 1.** [3] *For any seed set S and node v, the probability that an influence propagation process from S can activate v equals the probability that S overlaps an RR set for v.*

Based on this result, the *RIS* algorithm runs in two steps:

1. Generate random RR sets from $\mathcal{G}$ until a threshold on the total number of steps taken has been reached.
2. Consider the maximum coverage problem of selecting k nodes to cover the maximum number of RR sets generated. Use the standard greedy algorithm for the problem to derive a $(1-1/e)$-approximate solution S_k^*. Return S_k^* as the seed set to use for activation.

The rationale behind *RIS* is as follows: if a node u appears in a large number of RR sets it should have a high probability to activate many nodes under the IC model; hence, u's expected influence should be large. As such, we can think of the number of RR sets u appears in as an estimator for u's expected influence. By the same reasoning, if a size-k node set S_k^* covers most RR sets, then S_k^* is likely to have the maximum expected influence among all size-k node sets in $\mathcal{G}$ leading to a good solution to the IM problem. As shown in [30], Lemma 1 is the key result that underpins the approximation guarantees of *RIS*. The main contribution of Borgs et al. is an analysis of their proposed threshold-based approach: *RIS* generates RR sets until the total number of nodes and edges examined during the generation process reaches a pre-defined threshold Γ. Importantly, Γ must be set large enough to ensure a sufficient number of samples have been generated to provide a good estimator for expected influence. They show that when Γ is set to $\Theta((m+n)k\log n/\epsilon^2)$, *RIS* runs in near-optimal time $O((m+n)k\log n/\epsilon^2)$, and it returns a $(1-1/e-\epsilon)$-approximate solution to the IM problem with at least constant probability.

Due to the more complex dynamics involved in propagation under the MCIC model, adapting the reverse sampling approach to solve EIL is far from trivial.

4 New Definitions

In this section we introduce new definitions that are crucial to the development of our approach. In particular, we formalize the notion of *obstructed* nodes which is required to capture the necessary conditions for saving a node.

Identifying Saved Nodes

Given set A_L of vertices and (unweighted) directed graph $g \sim \mathcal{G}$, write $cl_g(A_L)$ for the set of nodes closer to A_L in G than to A_C in g. That is, a node $w \in cl_g(A_L)$ if there exists a node v such that $v \in A_L$ and $|SP_G(v,w)| \leq |SP_g(A_C,w)|$ where $SP_H(v,w)$ denotes a shortest path from node v to w in graph H and $SP_H(S,w)$ for a set S denotes the shortest path from any node $v \in S$ to w in graph H. When g is drawn from $\mathcal{G}$ this is a necessary, but not sufficient[1], condition for the set of nodes *saved* by A_L. We also require that the nodes in $cl_g(A_L)$ not be *obstructed* by the diffusion of campaign C in g.

[1] In Budak et al.'s work, the set of nodes closer to A_L than A_C is established as a necessary and sufficient condition to *save* a node in the MCIC model, but we note that this should be revised to include our *obstructed* condition due to a gap in the proof of Claim 1 in [4].

Table 1 Frequently used notation.

Notation	Description
$\mathcal{G}$	a social network represented as a weighted directed graph $\mathcal{G}$
G, G_T	the underlying unweighted graph G and its transpose G_T constructed by reversing the direction of each edge
g	a possible world for campaign C obtained by sampling each edge $e \in \mathcal{G}$ independently with probability $p_C(e)$
n, m	the number of nodes and edges in $\mathcal{G}$ respectively
k	the size of the seed set for misinformation prevention
C, L	the misinformation campaign C and the limiting campaign L
$p_C(e)$, $p_L(e)$	the propagation probability on an edge e for campaigns C and L respectively
$\pi(S)$	the prevention of a node set S in a misinformation propagation process on $\mathcal{G}$ (see Section 4)
$\omega(R)$, $\omega_\pi(R)$	the number of edges considered in generating an RRC set and that originate from nodes in an RRC set R (see Equation 3)
$\mathcal{R}$	the set of all RRC sets generated by Algorithm 1
$\mathcal{F}_\mathcal{R}(S)$	the fraction of RRC sets in $\mathcal{R}$ that are covered by a node set S
EPT	the expected width of a random RRC set
OPT_L	the maximum $\pi(S)$ for any size-k seed set S
λ	see Equation 4

▶ **Definition 3** (Obstructed Nodes). *A node $w \in cl_g(A_L)$ is* obstructed *and cannot be saved by A_L if for every path p from A_L to w there exists a node u on p such that $|SP_g(A_C, u)| < |SP_G(A_L, u)|$.*

Let $obs_g(A_L)$ be the set of obstructed nodes for A_L. Conceptually, the nodes in $obs_g(A_L)$ are cutoff because some node on the paths from A_L is reached by campaign C before L which stops the diffusion of L.

To help illustrate the concept of obstructed nodes, consider the graph presented in Figure 1 and the following possible world instance. Assume that the solid lines are *live* edges that make up the deterministic graph $g \sim \mathcal{G}$ for campaign C in the influence propagation process. The dashed lines are edges that were not realized for campaign C. The adversary campaign C starts from v_c while the limiting campaign L starts from v. Recall, the deterministic graph G for campaign L in this possible world instance is comprised of *both* the solid and dashed edges due to the high effectiveness property. Observe that $|SP_G(v, w)| = 4$ and $|SP_g(A_C, w)| = 5$. However, w cannot be saved in the resulting cascade since at timestamp 1 the node u will adopt campaign C. This intersects the shortest path from v to w and therefore campaign L will not be able to reach node w since a node never switches campaigns. Thus, we say that node w is *obstructed* by C.

Prevention & Saviours

Next, we formally define the prevention, $\pi(A_L)$, which corresponds to the number of nodes saved by A_L. That is, $\pi(A_L) = |R_g(A_C) \cap (cl_g(A_L) \setminus obs_g(A_L))|$ where $R_H(S)$ is the set of nodes in graph H that are *reachable* from set S (a node v in H is reachable from S if there exists a directed path in H that starts from a node in S and ends at v). We write $\mathbb{E}[\pi(A_L)] = \mathbb{E}_{g\sim\mathcal{G}}[\pi(A_L)]$ for the expected prevention of A_L in $\mathcal{G}$. Finally, let $OPT_L = max_{S:|S|=k}\{\mathbb{E}[\pi(S)]\}$ be the maximum expected prevention of a set of k nodes.

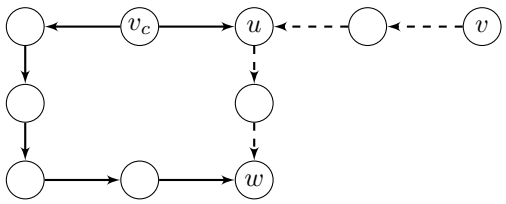

Figure 1 An example illustrating the concept of obstructed nodes where the possible world graph for campaign C is made up of the solid edges and the possible world for campaign L is made up of both solid and dashed lines.

We refer to the set of nodes that could have saved u as the *saviours* of u. A node w is a candidate saviour for u if there is a directed path from w to u in G (i.e. reverse reachability). Then, w is a saviour for u subject to the additional constraint that w would not be cutoff by the diffusion of A_C in g. That is, a candidate saviour w would be cutoff and cannot be a saviour for u if for every path p from w to u there exists a node v_b such that $|SP_g(A_C, v_b)| < |SP_G(w, v_b)|$. We refer to the set of candidate saviours for u that are cutoff as $\tau_g(u)$. Thus, we can define the saviours of u as the set $R_{G^T}(u) \setminus \tau_g(u)$. Therefore, we have:

▶ **Definition 4** (Reverse Reachability without Cutoff Set). *The reverse reachability without cutoff (RRC) set for a node v in $g \sim \mathcal{G}$ is the set of saviour nodes of v, i.e. the set of nodes that can save v. (That is, for each node u in the RRC set, $u \in R_{G^T}(u) \setminus \tau_g(u)$.) If $v \notin R_g(A_C)$ then we define the corresponding RRC set as empty since v is not eligible to be saved.*

▶ **Definition 5** (Random RRC Set). *A random RRC set is an RRC set generated on an instance of $g \sim \mathcal{G}$, for a node selected uniformly at random from g.*

Closing the Gap

Before presenting our reverse sampling approach, we make the following remark regarding obstruction in the context of prior work. The key observation that lead to our definition of obstructed nodes is that the shortest path condition must hold along the *entire* path. This observation was missed by [4] in the MCIC model. Instead, a correct *recursive* definition was provided for the set of nodes that are saved, but the resulting characterization based on shortest paths misses the crucial case of nodes that are obstructed.

Importantly, the solution in [4] can be recovered with a modified proof for Claim 1 and Theorem 4.2. In particular, the statements must include the notion of obstructed nodes in their *inoculation graph* definition, but a careful inspection shows that their objective function remains submodular after this inclusion. As a result, the greedy approach of [4] still provides the stated approximation guarantees and also allows us to incorporate the ideas of [3] in our solution (as [3] requires a submodular objective function as well).

5 Reverse Prevention Sampling

This section presents our misinformation prevention method, *Reverse Prevention Sampling (RPS)*. At a high level, *RPS*, in the same spirit as *RIS*, consists of two steps. In the first step it derives a parameter θ that ensures a solution of high quality will be produced. In the second step, using the estimate θ from step one, it generates θ RRC sets and then computes the maximum coverage on the resulting collection. More precisely, the two steps are:

1. **Parameter Estimation.** Compute a lower-bound for the maximum expected prevention among all possible size-k seed sets for A_L and then use the lower-bound to derive a parameter θ.
2. **Node Selection.** Sample θ random RRC sets from $\mathcal{G}$ to form a set $\mathcal{R}$ and then compute a size-k seed set S_k^* that covers a large number of RRC sets in $\mathcal{R}$. Return S_k^* as the final result.

In the rest of this section, we first tackle the challenging task of correctly generating RRC sets in the Node Selection step under the MCIC model. Next, we identify the conditions necessary for the Node Selection of *RPS* to return a solution of good quality and then describe how these conditions are achieved in the Parameter Estimation phase. Table 1 provides a reference to some of the frequently used notation. All proofs can be found in the full version [28].

Node Selection

The pseudocode of *RPS*'s Node Selection step is presented in Algorithm 1. Given $\mathcal{G}$, k, A_C, and a constant θ as input, the algorithm stochastically generates θ random RRC sets, accomplished by repeated invocation of the prevention of misinformation process, and inserts them into a set $\mathcal{R}$. Next, the algorithm follows a greedy approach for the *maximum coverage problem* to select the final seed set. In each iteration, the algorithm selects a node v_i that covers the largest number of RRC sets in $\mathcal{R}$, and then removes all those covered RRC sets from $\mathcal{R}$. The k selected nodes are put into a set S_k^*, which is returned as the final result.

Algorithm 1 NodeSelection($\mathcal{G}$,k,A_C,θ).

1: $\mathcal{R} \leftarrow \emptyset$
2: Generate θ random RRC sets and insert them into $\mathcal{R}$.
3: Initialize a node set $S_k^* \leftarrow \emptyset$
4: **for** $i = 1,\ldots,k$ **do**
5: Identify the node v_i that covers the most RRC sets in $\mathcal{R}$
6: Add v_i into S_k^*
7: Remove from $\mathcal{R}$ all RRC sets that are covered by v_i
8: **return** S_k^*

Lines 4-8 in Algorithm 1 correspond to a standard greedy approach for a *maximum coverage problem*. The problem is equivalent to maximizing a submodular function with cardinality constraints for which it is well known that a greedy approach returns a $(1-1/e)$-approximate solution in linear time [21].

5.1 RRC set generation

Next, we describe how to generate RRC sets correctly for the EIL problem under the MCIC model, which is more complicated than generating RR sets for the IC model [30]. The construction of RRC sets is done according to Definition 4. Recall that in the MCIC model, whether a node can be saved or not is based on a number of factors such as whether v is reachable via a path in $g \sim \mathcal{G}$ from A_C and the diffusion history of each campaign. Our algorithms tackle the complex interactions between campaigns by first identifying nodes that can be influenced by C which reveals important information for generating RRC sets for L.

Line 2 generates $\mathcal{R}$ by repeated simulation of the misinformation prevention process. The generation of each random RRC set is implemented as two breath-first searches (BFS) on $\mathcal{G}$ and G^T respectively. The first BFS is a *forward labelling* process from A_C implemented as a

forward BFS on $\mathcal{G}$ that computes the influence set of A_C in a possible world. The second BFS on G^T is a novel bounded-depth BFS with pruning that carefully tracks which nodes will become obstructed and is described in detail below.

Forward BFS with Lazy Sampling

We first describe the forward labelling process. As the forward labeling is unlikely to reach the whole graph, we simply reveal edge states on demand ("lazy sampling"), based on the principle of deferred decisions. Given the seed set A_C of campaign C, we perform a randomized BFS starting from A_C where each outgoing edge e in $\mathcal{G}$ is traversed with $p_C(e)$ probability. The set of nodes traversed in this manner ($R_g(A_C)$) is equivalent to $I(A_C)$ for $g \sim \mathcal{G}$, due to deferred randomness. Note that in each step of the above BFS we record at each node w the minimum distance from A_C to w, denoted $D(w)$, for use in the second BFS.

Given a randomly selected node u in G, observe that for u to be able to be saved we require $u \in R_g(A_C)$. Therefore, if the randomly selected node $u \notin R_g(A_C)$ then we return an empty RRC set. On the other hand, if $u \in R_g(A_C)$, we have $D(u) = |SP_g(A_C, u)|$ as a result of the above randomized BFS which indicates the maximum distance from u that candidate saviour nodes can exist. We run a second BFS from u in G^T to depth $D(u)$ to determine the saviour nodes for u by carefully pruning those nodes that would become obstructed.

Bounded-depth BFS with Pruning

The second BFS on G^T, presented in Algorithm 2, takes as input a source node u, the maximum depth $D(u)$, and a directed graph G^T. Algorithm 2 utilizes special indicator values associated with each node w to account for potential cutoffs from C. Each node w holds a variable, $\beta(w)$, which indicates the distance beyond w that the BFS can go before the diffusion would have been cutoff by C propagating in g. The β value for each node w is initialized to $D(w)$. In each round, the current node w has an opportunity to update the β value of each of its successors only if $\beta(w) > 0$. For each successor z of w, we assign $\beta(z) = \beta(w) - 1$ if $\beta(z) = \texttt{null}$ or if $\beta(z) > 0$ and $\beta(w) - 1 < \beta(z)$. In this way, each ancestor of z will have an opportunity to apply a β value to z to ensure that if any ancestor has a β value then so will z and furthermore, the β variable for z will be updated with the smallest β value from its ancestors. We terminate the BFS early if we reach a node w with $\beta(w) = 0$.

Figure 2 captures the primary scenarios encountered by Algorithm 2 when initialized at u. The enclosing dotted line represents the extent of the influence of campaign C for the current influence propagation process. First, notice that if the BFS moves away from $A_C = \{v_c\}$, as in the case of node z, that, once we move beyond the influence boundary of C, there will be no potential for cutoff. As such, the BFS is free to traverse until the maximum depth $D(u)$ is reached. On the other hand, if the BFS moves towards (or perpendicular to) v_c then we must carefully account for potential cutoff. For example, when the BFS reaches v, we know the distance from v_c to v: $D(v) = SP_g(v_c, v)$. Therefore, the BFS must track the fact that there cannot exist saviours at a distance $D(v)$ beyond v. In other words, if we imagine initializing a misinformation prevention process from a node w such that $SP_G(v, w) > D(v)$ then v will adopt campaign C before campaign L can reach v. Therefore, at each out-neighbour of v we use the knowledge of $D(v)$ to track the distance beyond v that saviours can exist. This updating process tracks the smallest such value and is allowed to cross the enclosing influence boundary of campaign C ensuring that all potential for cutoff is tracked.

Finally, we collect all nodes visited during the process (including u), and use them to form an RRC set. The runtime of this procedure is precisely the sum of the degrees (in G) of the nodes in $R_g(A_C)$ plus the sum of the degrees of the nodes in $R_{G^T}(u) \setminus \tau(u)$.

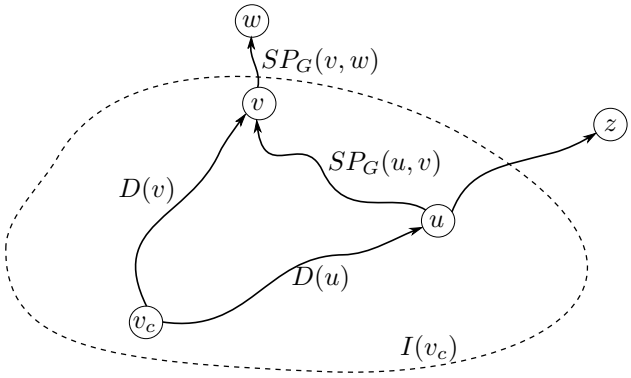

Figure 2 An overview of the primary scenarios encountered by Algorithm 2.

We briefly note another key difference between *RPS* and *RIS* occurs in the RRC set generation step. Unlike in the single campaign setting, generating an RRC set is comprised of two phases instead of just one. First, we are required to simulate the spread of misinformation since being influenced by campaign C is a pre-condition for being saved. As a result, only a fraction of the simulation steps of *RPS* provide signal for the prevention value we are trying to estimate. This difference is made concrete in the running time analysis to follow.

Algorithm 2 generateRRC(u, $D(u)$, G^T).

```
let R ← ∅, Q be a queue and Q.enqueue(u)
set u.depth = 0 and label u as discovered
while Q is not empty do
    w ← Q.dequeue(), R ← R ∪ {w}
    if w.depth = D(u) OR β(w) = 0 then
        continue
    for all nodes z in G^T.adjacentEdges(w) do
        if β(w) > 0 AND β(z) > 0 then
            if β(w) − 1 < β(z) then
                β(z) ← β(w) − 1
        else if β(w) > 0 then
            β(z) ← β(w) − 1
        if z is not labelled as discovered then
            set z.depth = w.depth + 1, label z as discovered and Q.enqueue(z)
return R
```

5.2 Analysis

In this section we focus on two parameters: solution quality and runtime. For Algorithm 1 to return a solution with approximation guarantee, we will provide a lower bound on θ. Then, we will analyze the running time of the algorithm in terms of θ and a quantity EPT that captures the expected number of edges traversed when generating a random RRC set.

Approximation Guarantee

We begin by establishing the crucial connection between RRC sets and the prevention process on $\mathcal{G}$. That is, the prevention of a set of nodes S is precisely n times the probability that a node u, chosen uniformly at random, has a saviour from S.

▶ **Lemma 2.** *For any seed set S and any node v, the probability that a prevention process from S can save v equals the probability that S overlaps an RRC set for v.*

For any node set S, let $F_{\mathcal{R}}(S)$ be the fraction of RRC sets in $\mathcal{R}$ covered by S. Then, based on Lemma 2, we can prove that the expected value of $n \cdot F_{\mathcal{R}}(S)$ equals the expected prevention of S in $\mathcal{G}$.

▶ **Corollary 1.** $\mathbb{E}[n \cdot F_{\mathcal{R}}(S)] = \mathbb{E}[\pi(S)]$

Corollary 1 implies that we can estimate $\mathbb{E}[\pi(S)]$ by estimating the fraction of RRC sets in $\mathcal{R}$ covered by S. The number of sets covered by a node v in $\mathcal{R}$ is precisely the number of times we observed that v was a saviour for a randomly selected node u. We can therefore think of $n \cdot F_{\mathcal{R}}(S)$ as an estimator for $\mathbb{E}[\pi(S)]$. Our primary task is to show that it is a *good* estimator. Using Chernoff bounds, we show that $n \cdot F_{\mathcal{R}}(S)$ is an accurate estimator of any node set S's expected prevention, when θ is sufficiently large:

▶ **Lemma 3.** *Suppose that θ satisfies*

$$\theta \geq (8 + 2\epsilon)n \cdot \frac{l \log n + \log \binom{n}{k} + \log 2}{OPT_L \cdot \epsilon^2} \tag{1}$$

Then, for any set S of at most k nodes, the following inequality holds with at least $1 - n^{-l}/\binom{n}{k}$ probability:

$$\left| n \cdot F_{\mathcal{R}}(S) - \mathbb{E}[\pi(S)] \right| < \frac{\epsilon}{2} \cdot OPT_L \tag{2}$$

Based on Lemma 3, we prove that if Eqn. 1 holds, Algorithm 1 returns a $(1 - 1/e - \epsilon)$-approximate solution with high probability by a simple application of Chernoff bounds.

▶ **Theorem 2.** *Given a θ that satisfies Equation 1, Algorithm 1 returns a $(1 - 1/e - \epsilon)$-approximate solution with at least $1 - n^{-l}$ probability.*

Runtime

First, we will define EPT which captures the expected number of edges traversed when generating a random RRC set. After that, we define the expected runtime of *RPS* in terms of EPT and the parameter θ.

Let M_R be the instance of $R_g(A_C)$ used in computing an RRC set R. Then, we define the *width* of an RRC set R, denoted as $\omega(R)$, as the number of edges in G that point to nodes in R plus the number of edges in G that originate from nodes in M_R. That is

$$\omega(R) = \sum_{u \in M_R} outdegree_G(u) + \sum_{v \in R} indegree_G(v) \tag{3}$$

Let EPT be the expected width of a random RRC set, where the expectation is taken over the randomness in R and M_R, and observe that Algorithm 1 has an expected runtime of $O(\theta \cdot EPT)$. This can be observed by noting that EPT captures the expected number of edge traversals required to generate a random RRC set since an edge is only considered in the propagation process (either of the two BFS's) if it points to a node in R or originates from a node in M_R.

An important consideration is that, since OPT_L is unknown, we cannot set θ directly from Equation 1. For simplicity, we define

$$\lambda = (8+2\epsilon)n \cdot \left(l \log n + \log \binom{n}{k} + \log 2\right) \cdot \epsilon^{-2} \tag{4}$$

and rewrite Equation 1 as $\theta \geq \lambda / OPT_L$. In the parameter estimation step we employ the techniques of [30] to derive a θ value for *RPS* that is above the threshold but also allows for practical efficiency.

5.3 Parameter Estimation

Our objective in this section is to identify a θ that makes $\theta \cdot EPT$ reasonably small, while still ensuring $\theta \geq \lambda / OPT_L$. We begin with some definitions. Let $\mathcal{V}^*$ be a probability distribution over the nodes in G, such that the probability mass for each node is proportional to its indegree in G. Let v^* be a random variable following $\mathcal{V}^*$ and recall that M_R is a random instance of $R_g(A_C)$ that is equivalent to the influence $I(A_C)$ for a possible world g. Furthermore, define $\omega(M_R)$, the number of edges in G that originate from nodes in M_R, as $\omega(M_R) = \sum_{u \in M_R} outdegree_G(u)$. Then we prove the following.

▶ **Lemma 4.** $\frac{m}{n} \cdot \mathbb{E}[\pi(\{v^*\})] = EPT - \mathbb{E}[\omega(M_R)]$, *where the expectation of* $\pi(\{v^*\})$ *and* $\omega(M_R)$ *is taken over the randomness in* v^* *and the prevention process.*

Lemma 4 shows that if we randomly sample a node from $\mathcal{V}^*$ and calculate its expected prevention p, then on average we have $p = \frac{n}{m}(EPT - \mathbb{E}[\omega(M_R)])$. This implies that $\frac{n}{m}(EPT - \mathbb{E}[\omega(M_R)]) \leq OPT_L$, since OPT_L is the maximum expected prevention of any size-k node set.

Recall that the expected runtime complexity of Algorithm 1 is $O(\theta \cdot EPT)$. Now, suppose we are able to identify a parameter t such that $t = \Omega(\frac{n}{m}(EPT - \mathbb{E}[\omega(M_R)]))$ and $t \leq OPT_L$. Then, by setting $\theta = \lambda / t$, we can guarantee that Algorithm 1 is correct, since $\theta \geq \lambda / OPT_L$, and has an expected runtime complexity of

$$O(\theta \cdot EPT) = O\Big(\frac{\lambda}{t} \cdot EPT\Big) = O\left(\frac{\lambda \cdot EPT}{\frac{n}{m}(EPT - \mathbb{E}[\omega(M_R)])}\right) \tag{5}$$

Furthermore, if we define a ratio $\gamma \in (0,1)$ which captures the relationship between $\mathbb{E}[\omega(M_R)]$ and EPT by writing $\mathbb{E}[\omega(M_R)] = \gamma EPT$, we can rewrite Equation 5 as

$$O\Big(\frac{m}{n}\Big(\frac{1}{1-\gamma}\Big)\lambda\Big) = O((k+l)(m+n)(1/(1-\gamma)) \log n / \epsilon^2) \tag{6}$$

Note that γ is a data-dependent approximation factor not present in [30], but arises from the MCIC model. In particular, the RRC set generation relies crucially on first computing the spread of misinformation from campaign C in order to determine the set of nodes that can be saved. See Section 6 for a detailed discussion of γ.

Computing t

We postpone the details of how to derive $t = \Omega(\frac{n}{m}(EPT - \mathbb{E}[\omega(M_R)]))$, a lower bound for the optimal prevention value, to the full version of the paper [28]. Briefly, we mimic the adaptive sampling approach of [30], which estimates a lower bound LB by dynamically adjusting the number of measurements based on the observed values of LB. The runtime required for the lower bound estimation is linear in Equation 6.

Wrapping Up

As a result, by Equation 6, *RPS* runs in $O((k+l)(m+n)(1/(1-\gamma))\log n/\epsilon^2)$ expected time. Furthermore, by Theorem 2 and the lower bound estimation, *RPS* returns a $(1-1/e-\epsilon)$-approximate solution with at least $1-3n^{-l}$ probability and the success probability can be increased to $1-n^{-l}$ by scaling l up by a factor of $1+\log 3/\log n$.

Finally, we note that the time complexity of *RPS* is *near-optimal* up to the instance-specific factor γ under the MCIC model, as it is only a $(\frac{1}{1-\gamma})\log n$ factor larger than the $\Omega(m+n)$ lower-bound proved in Section 6 (for fixed k, l, and ϵ).

6 Lower Bounds

Comparison with *MCGreedy*

MCGreedy runs in $O(kmnr)$ time, where r is the number of Monte Carlo samples used to estimate the expected prevention of each node set. Budak et al. do not provide a detailed analysis related to how r should be set to achieve a $(1-1/e-\epsilon)$-approximation ratio in the MCIC model, only pointing out that when each estimation of expected prevention has ϵ relative error, *MCGreedy* returns a $(1-1/e-\epsilon')$-approximate solution for a particular ϵ' [4]. In the following lemma, we present a more precise characterization of the relationship between r and *MCGreedy*'s approximation ratio in the MCIC model.

▶ **Lemma 5.** MCGreedy *returns a* $(1-1/e-\epsilon)$*-approximate solution with at least* $1-n^{-l}$ *probability, if*

$$r \geq (8k^2+2k\epsilon)\cdot n\cdot \frac{(l+1)\log n+\log k}{\epsilon^2\cdot OPT_L} \tag{7}$$

Assume that we know OPT_L in advance and set r to the smallest value satisfying the above inequality, in *MCGreedy*'s favour. In that case, the time complexity of *MCGreedy* is $O(k^3lmn^2\epsilon^{-2}\log n/OPT_L)$. Towards comparing *MCGreedy* to *RPS*, we show the following upper bound on the value of γ.

▶ **Claim 1.** $\gamma \leq \frac{n}{n+1}$

Claim 1 shows that the expected runtime for *RPS* is at most $O((k+l)mn\epsilon^{-2}\log n)$. As a consequence, given that $OPT_L \leq n$, the expected runtime of *MCGreedy* is always more than the expected runtime of *RPS*. In practice, we observe that for typical social networks $OPT_L \ll n$ and $\frac{1}{1-\gamma} \ll n+1$ resulting in superior scalability of *RPS* compared to *MCGreedy*.

A Lower Bound for EIL

In the theorem below, we provide a lower bound on the time it takes for any algorithm to compute a β-approximation for the EIL problem given uniform node sampling and an adjacency list representation. Thus, we rule out the possibility of a sublinear time algorithm for the EIL problem for an arbitrary β.

▶ **Theorem 3.** *Let* $0<\epsilon<\frac{1}{10e}$, $\beta \leq 1$ *be given. Any randomized algorithm for EIL that returns a set of seed nodes with approximation ratio* β*, with probability at least* $1-\frac{1}{e}-\epsilon$*, must have a runtime of at least* $\frac{\beta(m+n)}{24\min\{k,1/\beta\}}$.

7 Generalization to the Multi-Campaign Triggering Model

The *triggering model* is an influence propagation model that generalizes the IC and LT models. It assumes that each node v is associated with a triggering distribution $\mathcal{T}(v)$ over the power set of v's incoming neighbors. An influence propagation process under the triggering model works as follows: (1) for each node v, take a sample from $\mathcal{T}(v)$ and define the sample as the triggering set of v, then (2) at timestep 1 activate the seed set S, and (3) in subsequent timesteps, if an active node appears in the triggering set of v, then v becomes active. The propagation terminates when no more nodes can be activated.

We can define a *multi-campaign* version of the triggering model (MCT) that generalizes the MCIC model by associating each node with a *campaign-specific* triggering distribution $\mathcal{T}_Z(v)$ where $Z \in \{C, L\}$. The propagation process under MCT proceeds exactly as under MCIC with the exception that activation between rounds (step 2) is determined by $\mathcal{T}_C(v)$ and $\mathcal{T}_L(v)$. To the best of our knowledge, we are the first to formally define a multi-campaign version of the triggering model.

The key aspect of the MCIC model that enabled the existence of obstructed nodes is that the two campaigns are allowed to propagate along *different sets of edges* in a possible world X. This is exactly the intuition captured by the example in Figure 1 and is caused by L and C having separate propagation probabilities in $\mathcal{G}$. As a result, the campaigns traverse potentially unique graphs in X and results in the possibility of the obstruction of L by C. This observation holds under the more general setting of MCT due to the campaign-specific triggering sets and so the obstruction phenomenon exists under the MCT model.

Following the observations made in [30], our solutions can be easily extended to operate under the multi-campaign triggering (MCT) model with a modified high effectiveness property. Under MCT, the high effectiveness property asserts that $\mathcal{T}_L(v) = in(v)$ where $in(v)$ is the set of in-neighbours of v in G. Observe that Algorithm 1 does not rely on anything specific to the MCIC model, except a subroutine to generate random RRC sets. Thus, we can revise the definition of RRC sets to accommodate the MCT model.

Due to space constraints, we delay the details showing that this revised solution retains the performance guarantees of *RPS* under the MCT model to the full version of the paper. However, we note that all of the theoretical analysis of *RPS* is based on the Chernoff bounds and Lemma 2, without relying on any other results specific to the MCIC model. Therefore, once we establish an equivalent to Lemma 2, it is straightforward to combine it with the Chernoff bounds to show that, under the MCT model, *RPS* provides the same performance guarantees as in the case of the MCIC model. Thus, we have the following theorem:

▶ **Theorem 4.** *Under MCT,* RPS *runs in* $O((k+l)(m+n)(1/(1-\gamma))\log n/\epsilon^2)$ *expected time, and returns a* $(1-1/e-\epsilon)$*-approximate solution with at least* $1-3n^{-l}$ *probability.*

8 Summary of Experiments

The focus of our experiments is *algorithm efficiency* measured in runtime where our goal is to demonstrate the superior performance of *RPS* compared to *MCGreedy*. We observe that *RPS* provides a significant improvement of several orders of magnitude over *MCGreedy*. Further, we confirm that $\frac{1}{1-\gamma} \ll n+1$ on our small datasets which is strong evidence that *RPS* will outperform *MCGreedy* on typical social networks. Finally, we observe that the vast majority of the computation time is spent on generating the RRC sets for $\mathcal{R}$. A detailed experimental analysis and discussion is provided in the full version [28].

9 Conclusion & Future Work

In this work we presented *RPS*, a novel and scalable approach to the EIL problem. We showed the correctness and a detailed running-time analysis of our approach. Furthermore, we provided two lower bound results: one on the running-time requirement for any approach to solve the EIL problem and another on the number of Monte Carlo simulations required by *MCGreedy* to return a correct solution with high probability. As a result, the expected runtime of *RPS* is always less than the expected runtime of *MCGreedy*. Finally, we describe how our approach can be generalized to a multi-campaign triggering model. In future work we plan to investigate how to adapt our approach to a scenario where the source of the misinformation is only partially known.

References

1 Bob Abeshouse. Troll factories, bots and fake news: Inside the Wild West of social media, February 2018. URL: https://www.aljazeera.com/blogs/americas/2018/02/troll-factories-bots-fake-news-wild-west-social-media-180207061815575.html.

2 Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *WINE'07*, pages 306–311, Berlin, Heidelberg, 2007. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1781894.1781932.

3 Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. Influence Maximization in Social Networks: Towards an Optimal Algorithmic Solution. *CoRR*, abs/1212.0884, 2012. URL: http://arxiv.org/abs/1212.0884.

4 C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *WWW'11*, 2011.

5 Wei Chen, Laks V. S. Lakshmanan, and Carlos Castillo. *Information and Influence Propagation in Social Networks*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013. doi:10.2200/S00527ED1V01Y201308DTM037.

6 Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining*, pages 88–97. IEEE, 2010.

7 Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. Least cost rumor blocking in social networks. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 540–549. IEEE, 2013.

8 Qizhi Fang, Xin Chen, Qingqin Nong, Zongchao Zhang, Yongchang Cao, Yan Feng, Tao Sun, Suning Gong, and Ding-Zhu Du. General Rumor Blocking: An Efficient Random Algorithm with Martingale Approach. In *International Conference on Algorithmic Applications in Management*, pages 161–176. Springer, 2018.

9 Peter Foster. 'Bogus' AP tweet about explosion at the White House wipes billions off US markets, April 2018. URL: https://www.telegraph.co.uk/finance/markets/10013768/Bogus-AP-tweet-about-explosion-at-the-White-House-wipes-billions-off-US-markets.html.

10 Amit Goyal, Francesco Bonchi, Laks V. S. Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Netw. Analys. Mining*, 3(2):179–192, 2013. doi:10.1007/s13278-012-0062-z.

11 Chris Graham. YouTube employee's Twitter account hacked to spread fake news during attack, April 2018. URL: https://www.telegraph.co.uk/technology/2018/04/04/youtube-employees-twitter-account-hacked-spread-fake-news-attack/.

12 Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. Claimbuster: The first-ever end-to-end fact-checking system. *PVLDB*, 10(12):1945–1948, 2017.

13 Laura Hautala. Reddit was a misinformation hotspot in 2016 election, study says, December 2018. URL: https://www.cnet.com/news/reddit-election-misinformation-2016-research/.

14 Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474. SIAM, 2012.

15 Kyomin Jung, Wooram Heo, and Wei Chen. Irie: Scalable and robust influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 918–923. IEEE, 2012.

16 David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, 2003. doi:10.1145/956750.956769.

17 Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *WSDM*, pages 324–332. ACM, 2018.

18 Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW '10*, pages 641–650, New York, NY, USA, 2010. ACM. doi:10.1145/1772690.1772756.

19 Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence Diffusion Dynamics and Influence Maximization in Social Networks with Friend and Foe Relationships. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 657–666, New York, NY, USA, 2013. ACM. doi:10.1145/2433396.2433478.

20 Yishi Lin and John CS Lui. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation*, 91:187–204, 2015.

21 G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978. doi:10.1007/BF01588971.

22 Nam P. Nguyen, Guanhua Yan, My T. Thai, and Stephan Eidenbenz. Containment of Misinformation Spread in Online Social Networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 213–222, New York, NY, USA, 2012. ACM. doi:10.1145/2380718.2380746.

23 Maya Oppenheim. YouTube shooting: Twitter and Facebook explodes with misinformation and hoaxes, April 2018. URL: https://www.independent.co.uk/news/world/americas/youtube-shooting-fake-news-twitter-facebook-identity-illegal-immigrant-hoax-misinformation-a8287946.html.

24 Nishith Pathak, Arindam Banerjee, and Jaideep Srivastava. A generalized linear threshold model for multiple cascades. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 965–970. IEEE, 2010.

25 Kai Shu, H Russell Bernard, and Huan Liu. Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*, pages 43–65. Springer, 2019.

26 Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

27 Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. Clearing contamination in large networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1435–1448, 2016.

28 Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. Reverse Prevention Sampling for Misinformation Mitigation in Social Networks, 2018. arXiv:1807.01162.

29 Olivia Solon. Facebook's failure: did fake news and polarized politics get Trump elected?, November 2018. URL: https://www.theguardian.com/technology/2016/nov/10/facebook-fake-news-election-conspiracy-theories.

30 Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence Maximization: Near-optimal Time Complexity Meets Practical Efficiency. In *Proceedings of the 2014 ACM SIGMOD International*

Conference on Management of Data, SIGMOD '14, pages 75–86, New York, NY, USA, 2014. ACM. doi:10.1145/2588555.2593670.

31 Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. Fake news detection in social networks via crowd signals. In *WWW*, pages 517–524. WWW, 2018.

32 Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.