Contrasting the Spread of Misinformation in Online Social Networks

Marco Amoruso
University of Salerno, Italy
marco.amoruso91@gmail.com

Daniele Anello
University of Salerno, Italy
daniele.anello@gmail.com

Vincenzo Auletta
University of Salerno, Italy
auletta@unisa.it

Diodato Ferraioli
University of Salerno, Italy
dferraioli@unisa.it

ABSTRACT

The emergence of online social networks has revolutionized the way people seek and share information. Nowadays, popular online social sites as Twitter, Facebook and Google+ are among the major news sources as well as the most effective channels for viral marketing. However, these networks also became the most effective channel for spreading misinformation, accidentally or maliciously. The widespread diffusion of inaccurate information or fake news can lead to undesirable and severe consequences, such as widespread panic, libelous campaigns and conspiracies. In order to guarantee the trustworthiness of online social networks it is a crucial challenge to find effective strategies to contrast the spread of the misinformation in the network.

In this paper we concentrate our attention on two problems related to the diffusion of misinformation in social networks: identify the misinformation sources and limit its diffusion in the network. We consider a social network where some nodes have already been infected from misinformation. We first provide an heuristics to recognize the set of most probable sources of the infection. Then, we provide an heuristics to place a few monitors in some network nodes in order to control information diffused by the suspected nodes and block misinformation they injected in the network before it reaches a large part of the network.

To verify the quality and efficiency of our suggested solutions, we conduct experiments on several real-world networks. Empirical results indicate that our heuristics are among the most effective known in literature.

CCS Concepts

•**Networks** → **Network dynamics;** *Network economics;* Social media networks;

Keywords

Social Network, Spread of Misinformation, Independent Cascade Model, Maximum Spanning Arborescence, Unbalanced Cut, Source Identification

Final Version. The final version of this paper is available at `https://www.ifaamas.org/Proceedings/aamas2017/pdfs/p1323.pdf`

1. INTRODUCTION

In recent years online social networks have revolutionized the way we communicate, seek and share information. Nowadays, millions of people use popular online social sites such as Twitter, Facebook and Google+ to publish, read, and spread information and they rely on these networks as their major news sources. The popularity of online social networks comes from their incredible efficiency in information dissemination and sharing based on trust relationships built among their users. However, these trust relationships can also be used to spread rumors, inaccurate or even fake information. Thus, online social networks became the most effective channel for spreading misinformation. Here, we consider as misinformation both the inaccurate and not verified information accidentally diffused by users, and fake information created and spread by malicious users to mislead people and obtain illicit profit. The diffusion of inaccurate information or fake news can lead to undesirable and severe consequences, such as widespread panic, libelous campaigns against competitors, conspiracies, frauds. For example, false rumors about an earthquake in Ghazni province in Afghanistan in 2012 caused thousands of people to leave their home for long time [7]. Similarly, a false rumor originated from Twitter in June 2011 about an injury suffered by the former U.S. President Obama caused a temporary instability in financial markets [14]. Several other cases of misinformation occurred recently, such as the diffusion of false information about vaccinations that is causing many parents to refrain from immunizing their children [18], or the panic created by tweets establishing that Ebola was rampant in US [20]. Threats related to misinformation on online social networks attracted attention of the scientific community [6, 27] and the problem of finding effective strategies to guarantee the trustworthiness of online social networks has been recognized as a priority [25]. In 2013 the World Economic Forum recognized the issue of "misinformation spread" has one of the top ten globally significant issues of the year [26].

The problem of contrasting the spread of misinformation in an online social network is complex and multi-faceted. We can identify three main steps: (i) recognize misinformation; (ii) identify misinformation sources; (iii) limit the diffusion of misinformation. In this paper we concentrate

our attention on the last two points. We consider a scenario where misinformation has already been diffused in the network and administrators have been able to recognize it and find the set of the infected users. We want to identify the sources of misinformation and limit their ability to continue in diffusing misinformation in the network.

Identifying its sources is crucial in contrasting misinformation, since it allows network administrators to understand the ultimate goals of the misinformation, recognize their targets, punish the guilty nodes or orchestrate effective strategies for containing its diffusion. Due to the size of online social networks, to recognize misinformation sources can be a very challenging task and in several scenarios it is not possible to identify them for certain. For this reason, social network administrators are reluctant to ban users from the network if they do not have incontrovertible evidence of their misbehavior. A more pragmatic approach is to create a list of "suspects" that can be monitored in order to recognize the misinformation that they could inject in the network in an early stage and thus reduce its effect.

The control can be performed through monitors placed on users to parse all their activities, recognize misinformation and block it. Monitors in social media as Twitter or Facebook, could be implemented by a computer-aided accounts whose duty is to recognize spam and other malicious information, and to execute accurate fact checking in order to validate information that goes through them. Monitoring could be also realized by real users that will be paid for false or malicious information that they recognize and block.

Given the huge number of users in online social networks it may be impossible or too expensive to place monitors on all users. On the other hand, it could be impossible or undesired to place monitors directly on suspect users, because we cannot have access to them or we do not want to raise their suspicions. Thus, we have to select a set of users to monitor, distinct from the set of suspected sources, such that we can guarantee misinformation injected into the network will be intercepted. Clearly, we cannot guarantee that misinformation is recognized as soon as it is created, but we would like to take the number of users exposed to misinformation small. Moreover, in several contexts we could have specific users that must be protected from misinformation. For example, we want to protect young teenagers from misinformation about violence.

Our Contribution. We model a social network as a directed weighted graph $G = (V, E, w)$, where V is the set of nodes in the network, $E \subseteq V \times V$ is the set of directed edges, and $w\colon E \to [0, 1]$ defines for each edge (u, v) the probability that u will transmit its information to v.

We model the diffusion of (mis-)information on this network through the *Independent Cascade Model*, that has first investigated by Goldenberg et al. [10, 11] and by Kempe et al. [15]. Given a set $S \subseteq V$ of sources, let $A(t)$ denote the set of nodes that have been infected at time t and $A_t = \bigcup_{t'<t} A(t')$ be the set of all nodes that have been infected within time t. Then, the process works in steps as follows. We start with $A(0) = S$. At each time step t, for every $u \in A(t-1)$ and every $v \in V \setminus A_{t-1}$ if the edge (u, v) exists then v will be infected with probability $w(u, v)$. The process ends after t^* steps if no new node is infected in step t^* (i.e. $A(t^*) = \emptyset$). In the following we will omit the subscript t when it is clear from the context.

In this paper we consider two optimization problems:

Source Identification (SI) problem: given the graph G and the set A of infected nodes, find a set S of nodes having maximum probability to be the sources of the infection;

Monitor Placement (MP) problem: given G, the set S of source nodes, the (possibly empty) set T of target nodes and integer k, find the minimal set M of nodes, disjoint from $S \cup T$, that is a cut of the graph completely separating S and T such that the side of the graph containing S has at most k nodes.

For both these problems we propose heuristic solutions. Our heuristics build on graph-theoretic background. We reduce the SI problem to the Maximum Spanning Arborescence/Branching problem, and our heuristics will be based on the algorithms proposed in [3, 5] and [2]. The core of our heuristic for the MP problem, instead, lies on the computation of a k–unbalanced cut [12].

We remark that we concentrate our attention on heuristics since both the problems are provably hard. Indeed, Lappas et al. [16] proved that, even if the number of sources k is known, the Source Identification problem is NP-hard to solve even on Direct Acyclic Graph and it is NP-hard even to produce a β-approximation for this problem, for every $\beta > 1$. Zhang et al. [28], instead, considered a slightly different version of our Monitor Placement problem and proved it is $\#P$-complete. It is easy to prove that the problem in [28] can be polynomially reduced to our MP problem and so also MP is $\#P$-complete.

To verify the quality and efficiency of our heuristics we conduct extensive experiments on three real-world networks: Gnutella, Wiki-Vote and Epinions. The results indicate that our heuristics sensibly outperform the most effective alternatives known in literature.

Related works. In recent years the issue of the spread of misinformation in social media received great attention not only by social and computer scientists but also by reporters, economists, social media businessman and politicians. In particular, research concentrated on three directions: how to model the diffusion of misinformation, how to distinguish misinformation from true information and how to limit its spread in the network. Here, we only refer to works in the third direction, that is the more relevant to this paper.

The first works on the source identification problem used simple epidemics models: that is, they describe the information diffusion process as an infection disease spreading over the population. These works adopted centrality measures to identify the sources of the diffusion process. In particular, Comin and da Fontoura Costa [4] run several experiments to compare degree, betweenness, closeness, and eigenvector centralities in identifying the sources of the misinformation.

Along the same line of research, Shah and Zaman [24] proposed a new centrality measure, named *rumor centrality*, and showed that it outperforms all the previously considered centrality measures. Rumor centrality revealed to be very influential and it has been largely used, and extensions and generalizations have been proposed to identify sources of epidemics spread in several different settings, varying in the number of sources, the topology of the network and the coarseness of information about the set of affected nodes that is known to the algorithm. We refer interested readers to the survey of Jiang et al. [13] and references therein.

Epidemics models assume that there exists a global parameter that describes the probability that a user is infected by a neighbor. While this assumption simplifies the compu-

tational complexity of the model, it fails in describing real-world situations where users are differently bent to accept information from their neighbors. To overcome this difficulty, the Independent Cascade model has been proposed as a generalization of the epidemics model where each edge has its specific activation probability. Clearly, this generalization makes the problem extremely more complex to deal with. Indeed, as discussed above, Lappas et al. [16] prove that for the Source Identification problem with an Independent Cascade model of diffusion it is NP-hard even to produce a β-approximation, for every $\beta > 1$.

This hardness result leaves us only two possible research directions: either we focus on special network topologies or we consider general heuristics with good experimental performances. Lappas et al. in [16] follow the first direction and study the Source Identification problem on tree networks.

Nguyen et al. [21], instead, follow the second direction and propose efficient heuristics for identifying sources of misinformation in general networks. In this work we build upon the contribution of [21]. We present a new heuristic approach whose performance turns out to be much better than algorithms previously presented. Moreover, we remark that both algorithms in [16] and [21] need to know in advance the number k of sources to find. Our heuristic, instead, works well even if the number of sources is not known.

A correct identification of sources can be very useful even for limiting the diffusion of misinformation. Two main approaches have been proposed in literature to address this problem. The first one, proposed by Budak et al. [1], requires that a true information campaign is initiated from a subset of highly influential nodes. In this way, the diffusion of misinformation and true information proceeds in parallel, except that nodes that have received the true information will be immune to the misinformation and will not transmit it. However, in order to have a true information campaign that would be effective in the tentative of limiting misinformation, one must carefully choose the seeds from which the diffusion starts. We remark that this approach requires perfect knowledge of the sources of misinformation in order to correctly selecting the seeds of the contrasting campaign.

Budak et al. [1] studied the computational complexity of this problem and proposed some preliminary solutions. A similar approach has been taken by Nguyen et al. in [22] and [19]. They introduced the Node Protector problem which aims to find the smallest set of highly influential nodes whose decontamination with good information helps to contain the viral spread of misinformation. They give inapproximability results and propose greedy approximation algorithms. Variants of the problem have been also considered by Fan et al. [8], that focused on a community-based network, and Zhang et al. [29] that, instead, not only aim to minimize the spread of misinformation, but also to maximize the diffusion of true information.

Zhang et al. [28] recently proposed a different approach for limiting the spread of misinformation. Namely, they propose to place monitors over the network that are able to detect misinformation and block it. A good monitor placement should satisfy two requirements: on one side, we would like to place as few monitors as possible, on the other side, we would like that our monitors limit the number of nodes exposed to misinformation. These two discording goals make the problem very difficult. Indeed, Zhang et al. [28] proved that the problem is $\#P$-complete, and proposed an heuristics for placing monitors so that misinformation is detected with high probability before it reaches target nodes.

In this work, we strengthen the model [28] by putting more stringent requirements on the number of nodes exposed to misinformation and requiring that misinformation is always detected (more details in Section 3). Nevertheless, experiments show that our heuristics has performance comparable or even better than the algorithm proposed in [28].

2. SOURCE IDENTIFICATION

We start by recalling the statement of the *Source Identification* problem. Here we are given a social network $G = (V, E, w)$ and a set $A \subseteq V$ of nodes infected by misinformation. We assume that misinformation diffused in G according to the Independent Cascade model starting from a number k (maybe unknown) of sources. Our goal is to discover the sources of the misinformation.

To this aim, we consider the subgraph H_A of G induced by A and w.l.o.g. we assume H_A is connected. In the following we will omit the subscript when it is clear from the context.

Our approach is built on the idea that the structure of the network H can help to guess how the misinformation diffused. In particular, we would like to find the most probable path that misinformation went through, conditioned on the fact that the set of infected nodes is A. We now discuss how we implement this idea and how we use it to compute a set of probable sources.

Warm-up: Single Source. Consider first the simpler case in which the misinformation starts from a unique source.

Our approach is based on the concept of *maximum spanning arborescence*. An arborescence of the graph G is a directed subgraph T on a subset $V' \subseteq V$ of vertices of G, such that there is a distinguished node $r \in V'$, called *root*, and a single directed path from r to every other vertex in V'. A *spanning arborescence* of G is an arborescence containing all the vertices of G. Roughly speaking, an arborescence is a directed tree and a spanning arborescence is a directed spanning tree. The weight of an arborescence $T = (V', E')$ is the sum of the weights of the edges in T, i.e., $W(T) = \sum_{(u,v)\in E'} w(u, v)$. The *maximum spanning arborescence* is a spanning arborescence of maximum weight.

Let T be a spanning arborescence of the subgraph H induced by the set of infected nodes A and let r be its root. We denote as $E_{r,T}$ the event that *misinformation spreads from r according to T*, i.e., if it occurs that in an information diffusion process starting from r and proceeding according to the Independent Cascade model each node $v \in H$ is infected by its unique predecessor in T.

Let $\mathcal{T}$ be the set of all the spanning arborescences of H and let $T^\star \in \mathcal{T}$ be a maximum spanning arborescence of H. It is immediate to see that the following observation holds.

OBSERVATION 1. $T^\star = \arg\max_{T\in\mathcal{T}} \Pr(E_{r,T})$.

Consider indeed a spanning arborescence T with root r and assume that misinformation spreads from r according to T. Then, at time 0 the root r is the unique infected node and at the next time step only its children in T become infected. Since in the Independent Cascade model each neighbor is infected independently, then the probability that all the children of r are infected at time step 1 is

$$\sum_{(r,v)\in E(T)} \Pr(r \text{ infects } v \mid r \text{ is infected}) = \sum_{(r,v)\in E(T)} p(r, v).$$

By recursively repeating this argument on all the levels of the arborescence T, and considering that for each node v of H there exists a unique path in T from r to v, we can prove that the probability that misinformation spreads from r according to T is equal to $W(T)$. Thus, $T^\star$ is the most probable spanning arboresence that misinformation went through to infect nodes in A.

Notice that the probability that a node is the source of the misinformation is the sum of the probabilities of all the arborescences rooted in that node. But computing this probability is not computationally affordable. However, the root of the maximum spanning arborescence T^* is a natural candidate to maximize this probability.

This simple observation then suggests a heuristic for identifying the single source of misinformation: to choose the root of the maximum spanning arborescence of the subgraph H induced by the set of infected nodes A.

Despite the simplifying assumptions, our approach has some very interesting features. First, the problem of computing spanning arborescences is a very well studied and a lot of algorithms are known both for general networks and for specific classes of graphs. In particular, it is possible to efficiently compute the maximum spanning arborescence of a graph through the *Chu-Liu/Edmonds algorithm* independently proposed by Chu and Liu [3] and by Edmonds [5]. Moreover, even if the approach is so simple, it turns out to perform very well in practice. In fact, in all the experiments we run (see Section 4 for more details) our heurstic was able to find the right source of misinformation in more than 70% of the cases, largely outperforming perfomance of the algorithm proposed in [21].

Multiple Sources. Clearly, the assumption that misinformation originated in only one source is too restrictive and in this paragraph we show how to relax it.

The heuristic proposed for the single source case appears to be hard to extend to the case of multiple sources because it is based on spanning arborescences. Thus, if we assume that misinformation diffuses along the edges of an arborescence it is not clear how to select sources out of the root of the arborescence. For example, if we select nodes that are close to the root, we are implicitly limiting the influence of the root node, but nodes that are far away from the root may be scarcely influential.

However, the idea on which the heuristic for single source is based can still be fruitful. Suppose that misinformation starts from k different sources and proceeds as in k parallel threads. Then we can model the diffusion process by simply considering multiple arborescences, up to one for each source. Hence, if we can identify these diffusion trees, we can choose their roots as natural candidates for misinformation sources. This motivates us to use branchings in places of arborescences.

A *branching* of the graph G is a forest of disjoint arborescences. In a natural way, we can define the *maximum spanning branching* of G as a set of disjoint arborescenses containing all the vertices of G and such that the sum of their edges' weights is maximum.

Following the approach described for the single source case, our heuristic computes a maximum spanning branching B for the subgraph H induced by the set A of infected nodes and then take the sources of the arborescenses in B as sources of misinformation. As for the case of arborescenses, algorithms are known to efficently computing a maximum spanning branching (see, e.g., [2]).

We would like to highlight that our approach does not need to know the number of sources that must be identified. In fact, it simply returns all the roots of the arborescenses in the branching, regardless of their number.

Fixed Number of Sources. Suppose now that the number of sources (or a bound to it) is given. Notice that this is the problem studied in [21] and [16].

It is natural to ask if our approach can work also in this case. Interestingly, the algorithm developed by Camerini et al. [2] returns not only a maximal spanning branching, but it also allows us to easily compute the next optimal branchings through local transformations. Formally, once that the best branching is given, one can compute the next maximal branching by swapping a single edge in the branching with a new edge that is actually not in the branching.

This property of the algorithm of Camerini et al. [2] suggests the following approach for computing a given number of misinformation sources: we compute a maximal spanning branching and, if it has a number of arborescenses different from k, continue to transform it and compute next best branchings as long as the algorithm produces a branching with exactly k roots.

We remark that this approach is consistent with the idea adopted in previous cases: the returned branching is the most probable set of arborescenses that model the diffusion of misinformation from exactly k sources, and the roots of its arborescenses are natural candidates as misinformation sources. However, this approach has the drawback to be potentially very time consuming. In fact, the property that the next best branching differs from the previous one in exactly one edge implies that it frequently occurs that the next best branching will have the same roots as the previous one. Hence, the number of changes that one need to make before a branching with exactly k roots is found can be very large.

To address this issue, we adopt a different approach: let S be the set of candidate sources we already found. We distinguish two cases. If $|S| < k$, then we first construct graph H' from H by removing all the nodes in S and their adjacent edges and then compute a new maximum spanning branching for H'. Then, we add the roots of this branching to S and iterate until we obtain at least k sources.

If, instead, $|S| > k$ (either after the first branching computation or after the addition of the roots of a newly computed branchings), we proceed as follows: order the arborescenses of the branching computed in the last iteration in non-increasing order by their weight and take the roots of the first $k - \ell$ arborescenses, where ℓ is the number of sources found in previous iterations.

As for the single source case, we run extensive experiments on our heuristic for the identification of multiple sources. In particular, in order to compare our heuristic with other algorithms proposed in literature we concentrated on the case of a fixed number of sources. We run experiments for 2, 3 and 4 sources that show how our heuristic largely outperforms [21]: in almost all the instances, our heuristic was able to identify at least half of the sources and in more than 40% of the instances it was able to identify all the sources.

3. MONITOR PLACEMENT

Having identified the misinformation sources we can now consider the problem of limiting their capacity to continue

in diffusing misinformation.

In [28] Zhang et al. suggest to use monitors to limit the spread of misinformation originated from a set of known sources. The role of these nodes should be to filter the information they receive and block what they recognize as misinformation. Their goal is to use as few monitors as possible and place them as close to the sources as possible to limit the number of nodes reached by misinformation.

Specifically, Zhang et al. [28] considered the following problem, named τ-Monitor Placement. Let G be a network and let S be a set of misinformation sources and t be a target node that we have to protect from misinformation. For every path l in the network, we denote by p_l the probability that (mis)information is transmitted along this path, i.e., $p_l = \prod_{(u,v)\in l} p(u,v)$. For a set L of paths we denote by L^* the event that for every path $\ell \in L$, the (mis)information does not go through every node of L. A set of monitors placed in a set $M \subseteq V$ of vertices detects misinformation if there is at least one path from a node in S to a node in M on which there is a successful diffusion. Formally, if $L_{S,M}$ denotes the set of all paths whose starting endpoints are in S and the final endpoints are in M, then the probability that the misinformation is detected turns out to be exactly $D(S,M) = 1 - \Pr(L^*_{S,M})$.

The τ-Monitor Placement problem asks for a subset M of vertices chosen among the vertices at distance at most δ from S, such that $t \notin L_{S,M}$ and the misinformation detection probability $D(S,M) \geq 1-\tau$. Zhang et al. proved that the τ-Monitor Placement problem is $\#P$-complete and presented a heuristic to compute a monitor placement. Their solution is based on the computation of a cut of the graph.

In this paper we consider a generalization of the τ-Monitor Placement problem, called the MP problem. The extension we consider is multifold. Indeed, we assume to have a set T of target nodes to protect from misinformation and we require that whenever misinformation spreads over the network starting from the known set S of sources, then it will be detected and blocked by monitors in M before it reaches nodes in T, i.e. $D(S,M) = 1$. Moreover, in order to limit more effectively the spread of misinformation, we put an explicit bound on the number of nodes that can receive misinformation before it is blocked by monitors. Specifically, we require that the number of nodes in $V \setminus (S \cup M)$ that lie in paths in $L_{S,M}$ is upper bounded by a parameter k. This requirement generalizes and strengthens the request in [28] of placing monitors in nodes within distance at most δ from S: if the number of nodes close to S is small our requirement achieves the same effect as the τ-Monitor Placement problem, but it allows to keep low the number of infected nodes even if there are many nodes around the sources.

Clearly, the hardness result for the τ-Monitor Placement problem given in [28] extends to our problem. Moreover, since our problem is much more constrained than the τ-Monitor Placement problem, we should expect that more monitors will be required and their placement would be more difficult to compute. However, we next propose an heuristic for the MP problem and we show that its performances are comparable, and in some cases even better, to [28] both in terms of number of monitors and of computation time.

Monitors and Cuts. Let us start by considering a simple setting where we have a network represented by the graph $G = (V, E, w)$, with $w(u,v) = 1$ for each edge $(u,v) \in E$, a single source s of misinformation and a single target t to protect. Let C be a (s,t)-cut of the graph G. By definition of cut, if we remove from G all edges in C then there will be no paths from s to t. Thus, by placing monitors in the endpoints of the edges in C we can guarantee that all the information diffused by s will be blocked before it reaches t.

Observe that the number of monitors required by this approach depends on the size of the cut. Then, to reduce the number of required monitors we need an $(s-t)$-cut of minimum size. However, our requirements are not only to protect t from the misinformation but also to have a small number of nodes exposed to misinformation. Observe that a minimum cut does not give any guarantee on the number of nodes that can be reached by the misinformation before monitors detect it. Suppose, for example, that the minimum cut contains only edges adjacent to t. In this case, by placing monitors on the endpoints of these edges we have that only the target node t and the nodes hosting the monitors are protected by the misinformation. Thus, we have to impose another constraint to our cut: the set of nodes reachable from s after the removal of the edges in the cut must be small. To meet these additional requirement we will consider *unbalanced cuts*.

Formally, given a graph G, a source s, a target t, and an integer k, a *k–unbalanced (s,t)-cut* is a partition of the nodes of the graph in two sets, L and R, such that $s \in L$, $t \in R$, and $|L| \leq k$. The size of the cut (L,R) is given by the number of edges that have an endpoint in L and the other endpoint in R, i.e. $W(L,R) = |\{(u,v) \in E\colon u \in L, v \in R\}|$. A minimum k–unbalanced (s,t) cut is a cut (L^*,R^*) such that $W(L^*,R^*) = \min_{L,R\colon s\in L, t\in R, |L|\leq k} W(L,R)$. Roughly speaking, a minimum k–unbalanced (s,t)-cut is a (s,t)-cut of minimum size among all the $(s-t)$-cuts where the source side is bounded to contain at most k nodes.

Interestingly, a polynomial time algorithm is known for computing a minimum k–unbalanced cut for every graph G [9, 12]. The basic idea of this algorithm consists in finding a minimum cut in a graph G^α obtained from G by adding edges of weight α from all the nodes of the graph to t. Clearly, if $\alpha = 0$ then $G^\alpha = G$. If $\alpha > 0$, instead, the size of a cut (L,R) of G^α is given by the size of the same cut in the original graph G plus an additive factor of $\alpha|L|$. As α increases the size of L becomes more and more relevant with respect to the size of the cut. Hence, if α is sufficiently large, then a cut of G^α becomes a k–unbalanced cut of G.

Even if this algorithm seems to be very "expensive" in computational terms (we could compute a lot of cuts to find the correct value of α), Gallo et al. [9] proved that using the *parametric-flow technique* we can efficiently build a new cut on top of the previous one. Moreover, Gallo et al. [9] give a procedure to compute the next α that rapidly converges to a value that produces a minimum k–unbalanced cut.

The Heuristic. Even if the core of our solution is given by the computation of an unbalanced cut, as described above, there are still several aspects and optimizations that have to be addressed in designing our heuristic.

First of all, the approach described above was designed for a single source - single target scenario on an unweighted graph (actually, we assumed that all edge weights are equal). Here, we will explain how we can adapt our approach to many sources - many targets scenarios on weighted graphs. We address the problem of many sources and targets through a *source and target contraction*. Let $G = (V, E, w)$ be a weighted graph representing our network and let S be the

set of sources and T be the set of targets. Then we consider a new graph $G^* = (V^*, E^*, w^*)$ in which we contract all sources in a single node s^*, and all targets in a single node t^*, i.e., $V^* = (V \setminus (S \cup T)) \cup \{s^*, t^*\}$ and $E^* = \bigcup_{i=1}^{5} E_i^*$, where $E_1^* = \{(u,v)\colon u,v \in V^* \setminus \{s^*,t^*\}\}$, $E_2^* = \{(s^*,v)\colon (u,v) \in E \text{ and } u \in S\}$, $E_3^* = \{(u,s^*)\colon (u,v) \in E \text{ and } v \in S\}$, $E_4^* = \{(t^*,v)\colon (u,v) \in E \text{ and } u \in T\}$, and $E_5^* = \{(u,t^*)\colon (u,v) \in E \text{ and } v \in T\}$. As for the weights, we clearly set $w^*(u,v) = w(u,v)$ for every $(u,v) \in E_1^*$. For edges $(s^*,v) \in E_2^*$, let $C(v)$ be the set of sources that are connected with v in the original graph, i.e., $C(v) = \{s \in S\colon (s,v) \in E\}$. Then, $w^*(s^*,v) = 1 - \prod_{s \in C}(1 - w(s,v))$, that is the probability that at least one of the source nodes transmit the misinformation to v. Similarly, for edges $(u,s^\star) \in E_3^*$, let $C(u)$ be the set of sources at which u is connected in the original graph, i.e., $C(u) = \{s \in S\colon (u,s) \in E\}$. Then, $w^*(u,s^*) = 1 - \prod_{s \in C}(1 - w(s,v))$. A similar approach can be taken for edges in E_4^* and E_5^*.

The graph G^* has now a single source s^* and a single target t^*. Since this graph is weighted we need to specify which cuts we should compute. A natural choice would be to take minimum cuts (i.e. cuts that minimize the sum of the weights of their edges). However, since an edge weight represents the probability that information flows on that edge, placing monitors on the endpoints of a minimum cut would mean to place monitors on endpoints of edges where it is unlikely that the misinformation spreads. Monitoring these edges can then be a useless waste of resources.

We propose, instead, to place monitors on edges with large transmission probability. This indeed would also help in reducing the number of nodes infected by misinformation: in fact, not only the monitor placement guarantees that target nodes will not be reached by misinformation and there are no more than k nodes reached by misinformation, but it may be also the case the number of infected nodes is much less than k since edges between nodes in $L(S,M)$ have small transmission probabilities. In order to achieve this goal, we run the minimum k–unbalanced (s^*,t^*)-cut procedure on the graph $\hat{G} = (V^*, E^*, \hat{w})$, where edge weights are integers and they are inversely proportional to their weights in G^*. We observe that the use of integer weights has the positive side effect to make easier to compute the next α to use in the computation of the unbalanced cut.

Another optimization is related to the placement of monitors in the endpoints of the unbalanced cut's edges. In our informal discussion for the single source case we stated that monitors can be placed on all the endpoints of the cut's edges. However, it is clearly unnecessary to place monitors on all these nodes. Instead, we will use a more clever placement algorithm in order to reduce the number of monitors. Specifically, given a cut (L,R) of $\hat{G}$, where L is the side of the cut that contains s^*, we consider the unweighted graph $C = (W,F)$ induced by the edges of (L,R), i.e, $W = \{u \in L\colon (u,v) \in E^*, v \in R\} \cup \{v \in R\colon (u,v) \in E^*, u \in L\}$ and $F = \{(u,v) \in E^*\colon u \in L, v \in R\}$. Then, we compute a *minimum vertex cover* M of C and place monitors in all the nodes in M. Notice that, since C is a bipartite graph, it is possible to compute its minimum vertex cover in polynomial time (via a reduction to a problem of min cut/max flow).

Summarizing, our procedure works as described in Algorithm 1. Notice that our heuristic may place a monitor in s^*. In this case, we simply replace s^* with all its neighbors.

Input: Graph G, Sources S, Targets T, and integer k.
Output: Monitor vertices M.

$G^*, s^*, t^* = \texttt{SourceContraction}(G, S, T)$
$\hat{G} = \texttt{WeightConversion}(G^*)$
$(L,R) = \texttt{UnbalancedCut}(\hat{G}, s^*, t^*, k)$
$C = \texttt{BipartiteGraphFromCut}(L,R)$
$M = \texttt{VertexCover}(C)$
return M

Algorithm 1: Algorithm for monitoring placement

4. EXPERIMENTS

To validate our proposed heuristics and compare their performances to other known solutions we conducted extensive experiments on three real-world data sets: `Wiki-Vote`, `Gnutella08`, `Epinions`. All these data sets are available at [17] and differ with respect to size and density: `Wiki-Vote` is a dense network with of 7115 nodes and 103689 edges; `Gnutella08` has comparable size, but it is much sparser, since it has 6301 nodes, but only 20777 edges; finally, `Epinions` is a large network of 75879 nodes and 508837 edges.

All these networks are directed and unweighted. To run our experiments we need to define transmission probabilities for all their edges. For `Wiki-Vote` and `Gnutella08`, these probabilities have been generated uniformly at random in $[0,1]$. As for `Epinions`, we adopted the approach described by Richardson et al. [23]: to each node u, it has been assigned a quantity $\gamma_u \in [0,1]$ chosen according to a Gaussian distribution with mean 0.5 and standard deviation 0.25; then to an edge (u,v) it is assigned weight $w(u,v)$ uniformly chosen from $[\max\{\gamma_u + \gamma_v - 1, 0\}, \min\{\gamma_u - \gamma_v + 1, 1\}]$.

Source Identification. For sake of comparison with other known solutions we run our experiments with a fixed number k of sources, with k ranging from 1 to 4. For every graph, we first placed the k sources uniformly at random in the network and then run an Independent Cascade diffusion process starting from the k sources. In this way we obtained the set A of the infected nodes. Then the graph and the set of infected nodes were given in input to our heuristic.

In order to test the heuristic's performance with respect to the number of infected nodes we grouped our experiments in five groups, depending on the cardinality of A: $[100,250]$, $[500,650]$, $[1000,1200]$, $[1500,1700]$, and $[2100,2700]$. To force each test to be in one of these ranges, we choose a random integer i within that range, and we stop the cascade process as soon as i nodes have been infected by misinformation. For each of these experimental settings, i.e., for every graph, each value of k, and each range, the experiment has been repeated at least 15 times.

First, we tested our heuristic for single source identification. As you can see in Figure 1, it was able to find the right sources in approx. 70% of the experiments, with a slight decrease of the success rate only when the number of infected nodes is very large. As a matter of comparison, we note that the algorithm proposed by Nguyen et al. [21], run on the same inputs, finds the correct source in less than 10% of experiments, and it never finds the correct source when the number of infected nodes is within the range $[2100,2700]$.

We also evaluated the performances of our heuristic with multiple sources. In order to compare our approach to the previous proposals, we considered only the case in which the number of sources is known. Clearly, in this case an algo-

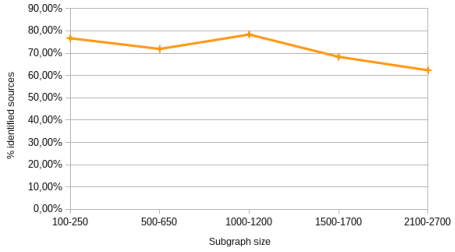


Figure 1: Success rate of the single source identification heuristic with respect to the number of infected nodes.

rithm can correctly identify all the sources or only part of them. Figure 2 shows the rate of (partial) successes of our heuristic when $k = 3$ and when $k = 4$ (results for $k = 2$ are similar and we do not present here). As you can see, in almost all the experiments our heuristic correctly identified at least half of sources and in more than 70% of experiments it correctly identified all sources except at most one. Moreover, it was able to correctly identify all the sources in at least 40% of experiments, even if the success rate tends to decrease as the number of infected nodes increases.

We remark that our success rate is more than five times larger than the one achieved on the same inputs by Nguyen et al. [21], and this rate is up to twenty times larger when the number of infected nodes is large.

Monitor Placement. To test performances of our heuristic for the MP problem we run experiments on `Wiki-Vote`, `Gnutella08` and on a subset of the `Epinions` network, named `SUB-Epinions`, consisting of 7479 nodes and 25855 edges. The network `SUB-Epinions` has been created by randomly choosing an integer n between 7000 and 7500, selecting n nodes at random from the largest strongly connected component of `Epinions`, and considering the graph induced by these nodes. Edge weights have been randomly assigned.

We decided to compare our heuristic with respect to the algorithm `MMSC` proposed in [28]. For this reason, we followed them in the choice of the sources of misinformation. We considered a set S of sources, with $|S| = 10, 20, \ldots, 50$, and only one target node t. Sources are selected randomly among the set of nodes with low outdegree that are neighbors of the $|S|$ nodes with the largest degrees. Target is selected uniformly at random among nodes with low indegree. Here, we say that the degree of node is low (high) if it is below (above) the average degree of the network.

For each graph G and each set of sources S, we first contracted sources in a single source (see Section 3 for details) and then we run algorithm `MMSC` with parameters $\tau = 0.1$ and $\delta \in \{1, 2\}$ (here δ denotes the maximum distance from the source at which it is possible to place the monitors, thus if we increase δ we are allowing more nodes to be infected by misinformation).

In order to make the results of the algorithms comparable, we would like to have more or less the same expected number of nodes that are reached by misinformation. For this reason, we run 100 separate executions of the Independent Cascade diffusion process on the network G with sources from S and monitors placed according to algorithm `MMSC`, and let k be the average number of nodes infected by misinformation in these executions. Then we run our heuristic on input (G, S, t, k)

For each graph, each value of $|S|$ and each value of δ we executed the experiment 10 times and evaluated both the average number of monitors and the average number of vertices reached by misinformation.

The results of our experiments show very different behaviors for the cases of $\delta = 1$ and $\delta = 2$. When $\delta = 1$ our heuristic places a number of monitors that is slightly greater than algorithm `MMSC`. In Figure 3a we show results only for the `Wiki-Vote` network but we had similar results also for the other two netowrks.

We remark that this slightly increase in the number of monitors, never greater than 20%, is counterbalanced by the much more stronger results of our heuristic in terms of limitations to the spread of misinformation. In fact, our heuristic guarantees complete protection of the target node from misinformation while `MMSC` allows that t could be reached by misinformation with small probability. Moreover, with our heuristic the average number of nodes that are reached by misinformation even in presence of monitors is much less than `MMSC` and the difference between the two algorithms explodes as the number of sources increases. As above, We show only the result for the `Wiki-Vote` network in Figure 3b, since results for `Gnutella08` and `SUB-Epinions` are similar.

When $\delta > 1$ our heuristic outperforms the `MMSC` algorithm with respect to both the number of monitors placed and he number of nodes exposed to the misinformation. As you can see in Figure 4a, the number of monitors placed by our heuristic remains almost unchanged regardless of the value of δ, whereas the number of monitors placed by `MMSC` explodes. Moreover, as shown in Figure 4b, even if `MMSC` places much more monitors, our heuristic has much better performances with respect to the number of nodes that can be reached by misinformation.

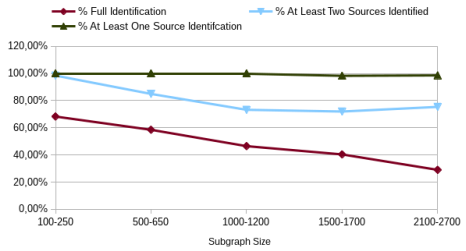
Finally, we compared running times of our heuristic and the `MMSC` algorithm to check if the much better performances of our heuristic could come at cost of a larger running time. We run our experiments on a CPU Intel Core i7 860 2.8 GHz, 4 core with 8MB cache and 4GB RAM. Figure 5a and Figure 5b show that the two algorithms run on `Wiki-Vote` have comparable running times for $\delta = 1$, but our heuristic becomes significantly faster when δ increases.

5. CONCLUSIONS AND FUTURE WORK

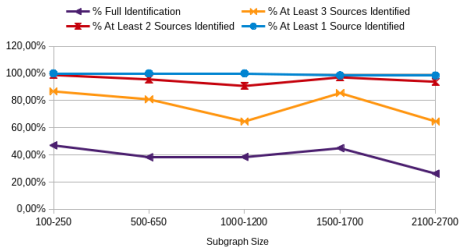
In this paper we considered the problem of contrasting the spread of misinformation in an online social network. We proposed two heuristics for first identifying the sources of misinformation and then placing a set of monitors on nodes of the network to limit the spread of misinformation.

Our heuristics are based on well-studied graph-theoretic algorithms, such as the algorithm for computing the maximum spanning branching of a directed graph, or the algorihtm to compute an unbalanced cut. Both our heuristics can have arbitrarily large approximation guarantees, due the previously known hardness results. However, they performed very well in the extensive tests we run on real-world networks and largely outperformed previously known algorithms.

As shown in the paper, the Monitor Placement heuristic obtains much better results while having comparable (in some cases even better) running times with previously known algorithms. Our solution to the Source Identification problem, instead, takes much more time than the previously

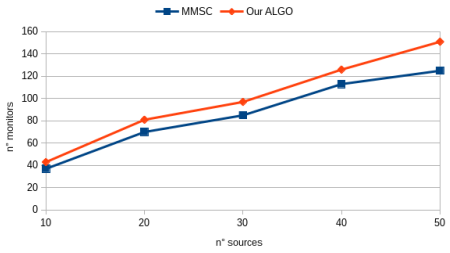


(a) $k = 3$

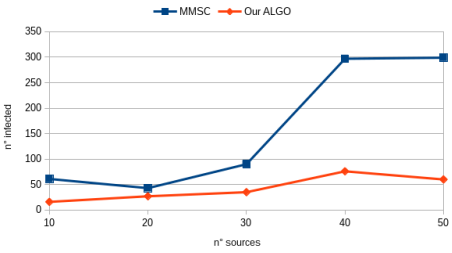


(b) $k = 4$

Figure 2: Success rate of the multiple source identification heuristic with respect to the number of infected nodes.

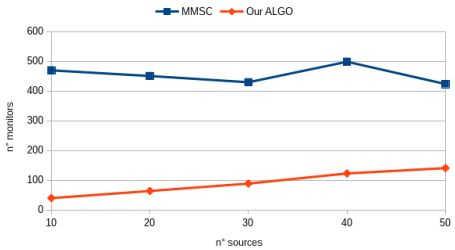


(a) Monitors

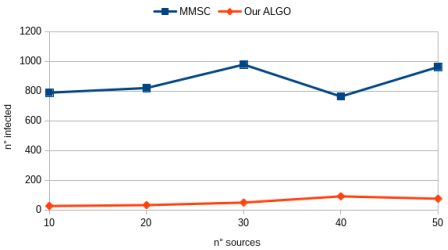


(b) Infected

Figure 3: Performances of the two algorithms for Monitor Placement on `Wiki-Vote`, when $\delta = 1$.

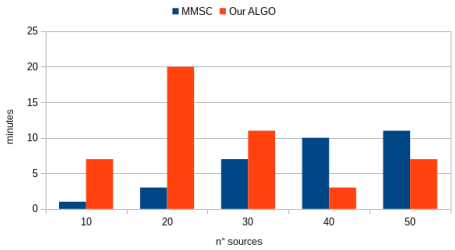


(a) Monitors

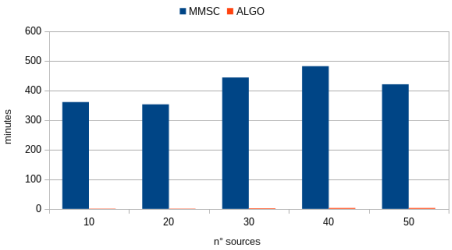


(b) Infected

Figure 4: Performance of the two algorithms for Monitor Placement on `Wiki-Vote`, when $\delta = 2$.



(a) $\delta = 1$



(b) $\delta = 2$

Figure 5: Running times of the two algorithms for monitor placement.

known heuristics. We believe that it would be interesting and useful to explore the possibility to either optimize our approach or design alternative and more efficient algorithms that achieve performances comparable with the ours in less time.

REFERENCES

[1] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674. ACM, 2011.

[2] P. M. Camerini, L. Fratta, and F. Maffioli. The k best spanning arborescences of a network. *Networks*, 10(2):91–109, 1980.

[3] Y.-J. Chu and T.-H. Liu. On the shortest arborescence of a directed graph. *Science Sinica*, 14, 1965.

[4] C. H. Comin and L. da Fontoura Costa. Identifying the starting point of a spreading process in complex networks. *Physical Review E*, 84(5):056105, 2011.

[5] J. Edmonds. Optimum Branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240, 1967.

[6] R. Ehrenberg. Social media sway: Worries over political misinformation on twitter attract scientists' attention. *Science News*, 182:22–25, 2012.

[7] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. 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] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. Bi. Least cost rumor blocking in social networks. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 540–549, 2013.

[9] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

[10] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001.

[11] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 2001:1, 2001.

[12] A. Hayrapetyan, D. Kempe, M. Pál, and Z. Svitkina. *Unbalanced Graph Cuts*, pages 191–202. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[13] J. Jiang, S. Wen, S. Yu, Y. Xiang, W. Zhou, and E. Hossain. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys and Tutorials, accepted*, 17(9), 2014.

[14] F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 8. ACM, 2013.

[15] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[16] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1059–1068. ACM, 2010.

[17] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[18] S. Lewandowsky, U. K. Ecker, C. M. Seifert, N. Schwarz, and J. Cook. Misinformation and its correction continued influence and successful debiasing. *Psychological Science in the Public Interest*, 13(3):106–131, 2012.

[19] S. Li, Y. Zhu, D. Li, D. Kim, and H. Huang. Rumor restriction in online social networks. In *2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC)*, pages 1–10. IEEE, 2013.

[20] V. Luckerson. Fear, misinformation, and social media complicate ebola fight, 2014.

[21] D. T. Nguyen, N. P. Nguyen, and M. T. Thai. Sources of misinformation in online social networks: Who to suspect? In *MILCOM 2012-2012 IEEE Military Communications Conference*, pages 1–6. IEEE, 2012.

[22] N. P. Nguyen, G. Yan, M. T. Thai, and S. Eidenbenz. Containment of misinformation spread in online social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 213–222. ACM, 2012.

[23] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *International semantic Web conference*, pages 351–368. Springer, 2003.

[24] D. Shah and T. Zaman. Rumors in a network: who's the culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, 2011.

[25] D. Talbot. Prevenitng misinformation from spreading through social media. *MIT Technology Review*, 2013.

[26] F. Vis. To tackle the spread of misinformation online we must first understand it. *Guardian Comment Network*, 2014.

[27] L. Wu, F. Morstatter, X. Hu, and H. Liu. Mining misinformation in social media. In M. T. Thai, W. Wu, and H. Xiong, editors, *Big Data in Complex and Social Networks*. Chapman and Hall/CRC, 2016.

[28] H. Zhang, M. A. Alim, M. T. Thai, and H. T. Nguyen. Monitor placement to timely detect misinformation in online social networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 1152–1157. IEEE, 2015.

[29] H. Zhang, H. Zhang, X. Li, and M. T. Thai. Limiting the spread of misinformation while effectively raising awareness in social networks. In *International Conference on Computational Social Networks*, pages 35–47. Springer, 2015.