Back to the Source:
An Online Approach for Sensor Placement and Source Localization

Brunella Spinelli
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
brunella.spinelli@epfl.ch

L. Elisa Celis
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
elisa.celis@epfl.ch

Patrick Thiran
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
patrick.thiran@epfl.ch

ABSTRACT

Abstract.

Source localization, the act of finding the originator of a disease or rumor in a network, has become an important problem in sociology and epidemiology. The localization is done using the infection state and time of infection of a few designated *sensor* nodes; however, maintaining sensors can be very costly in practice.

We propose the first online approach to source localization: We deploy *a priori* only a small number of sensors (which reveal if they are reached by an infection) and then iteratively choose the best location to place a new sensor in order to localize the source. This approach allows for source localization with a very small number of sensors; moreover, the source can be found while the epidemic is still ongoing. Our method applies to a general network topology and performs well even with random transmission delays.

Keywords

Epidemics; Sensor Placement; Online Source Localization

1. INTRODUCTION

Computer worms, or rumors spreading on social networks, often trigger the question of how to identify the source of an epidemic. This problem also arises in epidemiology, when health authorities investigate the origin of a disease outbreak. The problem of *source localization* has received considerable attention in the past few years; because of its combinatorial nature, it is inherently difficult: the infection of a few nodes can be explained by multiple and possibly very different epidemic propagations. Researchers have considered various models and algorithms that differ in the epidemic spreading model and in the information that is available for source localization. Such models are often not realistic, either because they rely on some strong assumptions about the epidemic features (tree networks, deterministic transmission delays, etc.) or because they require an overwhelming amount of information to localize the source.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052584

The costs of retrieving information for source localization cannot be disregarded. Data collection is never free; moreover, due to privacy concerns, individuals are becoming aware of the value of their data and resistant to share it for free [9]. In the case of infectious diseases, performing the necessary medical exams and the subsequent data analysis on many suspected households or communities can be exorbitantly expensive, whereas the efficient allocation of resources can lead to enormous savings [29].

Driven by the demand for general models for source localization and by practical resource-allocation constraints, we adopt a very general setting in terms of the epidemic model and prior information available, and we focus on designing a resource-efficient algorithm for information collection and source localization.

Our model. We model the connections across which an epidemic can spread with an undirected network $\mathcal{G}(V, E)$ of size $N = |V|$. Each edge $uv \in E$ is given a weight $w_{uv} \in \mathbb{R}^+$ that is the expected time required for an infection to spread from u to v. The edge weights induce a distance metric d on $\mathcal{G}$: $d(u, v)$ is the length of the shortest path from u to v.

An epidemic spreads on $\mathcal{G}$ starting from a single source $v^\star$ at an *unknown time* $t^\star$. The unknown source $v^\star$ is drawn from a prior distribution π on V. At any time, a node can be in one of two states: *susceptible* or *infected*. If u becomes infected at time t_u, a susceptible neighbor v of u will become infected at time $t_u + \theta_{uv}$, where θ_{uv} is a continuous random variable with expected value w_{uv}.

When a node is chosen as a *sensor*, it can reveal its infection state and, if it is infected, its infection time. We have two different types of sensors: *static* sensors $\mathcal{S}$ and *dynamic* sensors $\mathcal{D}$. Static sensors are placed *a priori* in the network. They serve the purpose of detecting any ongoing epidemic and of triggering the source search process. When a static sensor $s_0 \in \mathcal{S}$ gets infected, the epidemic is detected and the online placement of the dynamic sensors starts.

Our results. Most source-localization approaches assume that all available sensors are chosen *a priori*, independently of any particular epidemic instance, and, commonly, the source can be localized only after the epidemic spreads across the entire network. Instead, we propose a novel approach where we start the source-localization process as soon as an epidemic is detected and we place dynamic sensors actively while the epidemic spreads.

We approach the problem of source-localization asking the following question: *Who is the most informative individual, given our current knowledge about the ongoing epidemic?* Indeed, depending on the particular epidemic instance, the

infection time or the state of some individuals might be more informative than that of others, hence we want to observe them, i.e., to choose them as *sensors*.

Our methods are practical because they apply to *general graphs* and both *deterministic* and *non-deterministic* settings. We validate our results with extensive experiments on synthetic and real-world networks. We experimentally show that, when we have a limited budget for the dynamic sensors, we dramatically outperform a static strategy with the same budget – improving the success rate of finding the source from ∼5% to ∼75% of the time. Moreover, when we are unconstrained by a budget, we can localize the source with few sensors: Many purely-static approaches to sensor placement require a large fraction of the nodes to be sensors (e.g., $> 30\%$, see the discussion in Section 5), while our dynamic placement uses ∼3% on all topologies (see Figure 2). Intuitively, the reason for these dramatic improvements is the dual approach of using static and dynamic sensors: Once a static sensor is infected, it effectively cuts down the network to a region of size $N/|\mathcal{S}|$ that contains the source. Then, the $|\mathcal{D}|$ dynamic sensors only need to localize the source in this smaller network. Proving this formally would be an interesting direction for future work.

We focus on studying source localization and dynamic sensor placement, assuming that a set of static sensors is *given*. We consider two objectives: first, under budget-constraints for the number of sensors, we are interested in minimizing the uncertainty on the identity of the source (i.e., the number of nodes that, given the available observations, have a positive probability of being the source); second, when the budget for sensors is not limited, we want to minimize the number of sensors needed to exactly identify the source.

2. PRELIMINARIES

2.1 Model Assumptions

What we assume. We make the following assumptions.

(A.1) We assume that the network topology is known. This is a common assumption when studying source localization (see, e.g., [28, 1, 26, 27, 22]).

(A.2) We assume that, when a node is chosen as dynamic sensor, it reveals its *state* (healthy or infected). If it is infected, it also reveals the time at which it became infected. This is not a strong assumption because, by interviewing social-networks users (or, in the case of a disease, patients), the infection time of an individual can be retrieved [38].

What we do *not* assume. In order to obtain a tractable setting, much prior work has made assumptions which are not always feasible in practice and which we do not make. In particular, we do not make the following assumptions.

(B.1) Knowledge of the *state of all the nodes* at a given point in time. This might be prohibitively expensive when one should maintain a very large number of monitoring systems [40]. Instead, we detect the source based on the infection time of a very small set of nodes.

(B.2) Knowledge of the *time at which the epidemic starts*. This information is in most practical cases not available [15, 26]. Hence we do not make assumptions about the starting time of the epidemic.

(B.3) Observation of *multiple epidemics*. Observing multiple epidemics started by the same source certainly helps in its localization [26, 11]. In this work, we consider a single epidemic because we are interested in localizing the source *while* the epidemic spreads.

(B.4) A specific *class of network topologies*. A large part of the literature assumes tree topologies. Having a unique path between any two nodes makes source localization much easier [15]. Instead, our methods work on arbitrary graphs.

(B.5) *Deterministic or discretized transmission delays*. When the transmission delays are deterministic, given the position of the source, the epidemic is deterministic. Hence, if the source is unknown, tracking back its position becomes much easier [30]. Also, assuming that infection times are discrete is limiting and may result in a loss of important information [4]. We assume transmission delays to be randomly drawn from continuous distributions with bounded support, which include deterministic delays as a particular case and can, in practice, approximate unimodal distributions with unbounded support (e.g., Gaussians).

(B.6) A specific *epidemic model*. Our method only uses the time of first-infection of the sensors (no assumption on recovery or re-infection dynamics is made). Hence, it can be applied to most epidemic models, including the well known SIS or SIR (provided that nodes do not recover before infecting their neighbors).

2.2 Model Description and Notation

Sensor Placement. The set of static sensors is denoted by $\mathcal{S}$, with $|\mathcal{S}| = K_s$. Let $\tau_0 \in \mathbb{R}$ be the first time at which a subset of static sensors $\mathcal{S}_0 \subseteq \mathcal{S}$ are infected. At this time the placement of dynamic sensors starts. A new dynamic sensor is placed at each time $\tau_i = \tau_0 + i\delta$, $i \in \mathbb{N}^+$, where $\delta > 0$ is called the *placement delay*.

The i^{th} dynamic sensor, i.e., the one placed at time τ_i, is denoted by d_i. The set of dynamic sensors deployed in the network before or at step i is denoted by $\mathcal{D}_i$. The number of dynamic sensors is limited by a *budget* K_d, hence the maximum total number of sensors is $K_s + K_d$. If we do not have a limited budget for dynamic sensors, we trivially set $K_d = \infty$. We stop adding dynamic sensors when the source is localized or when the number of dynamic sensors reaches the budget K_d. The set of all static and dynamic sensors is denoted by $\mathcal{U}$. The cardinality of the latter set, $|\mathcal{U}|$, is the total number of sensors used in the localization process and is our metric for the *cost* of localization.

Positive and Negative Observations. A sensor gives information in two possible ways: If it is infected, it reveals its infection time; otherwise it reveals that it is susceptible. In the first (respectively, second) case we say that the sensor gives a *positive* (respectively, *negative*) *observation*. We will see that an observation contributes to the localization process even if it is negative. We represent each observation ω as a tuple (u_ω, t_ω) where $u_\omega \in V$ denotes the sensor and $t_\omega \in \mathbb{R}$ is its infection time if the observation is positive, whereas $t_\omega = \emptyset$ if the observation is negative. For every step i of the localization process, we denote the set of all observations (positive or negative) collected before or at time τ_i by $\mathcal{O}_i$. Specifically, $\mathcal{O}_0 = \{(s, \tau_0), s \in \mathcal{S}_0\} \cup \{(s, \emptyset), s \in \mathcal{S} \backslash \mathcal{S}_0\}$ and, for $i \in \mathbb{N}^+$, $\mathcal{O}_i \backslash \mathcal{O}_{i-1}$ contains the new observation of sensor d_i and the positive observations (if any) of the previously placed sensors that get infected in $(\tau_{i-1}, \tau_i]$. Denoting with $\mathcal{I}_i$ the set of nodes which become infected in $(\tau_{i-1}, \tau_i]$

Notation

$\mathbb{N}$ ($\mathbb{N}^+$)	positive integers including (excluding) 0
$\mathcal{G}(E,V)$	network
w_{uv}	weight of edge (u,v)
$\mathcal{S}$	set of static sensors
$\mathcal{D}$	set of dynamic sensors
$\mathcal{U}$	$\mathcal{S} \cup \mathcal{D}$
K_s	number of static sensors, $K_s = \lvert\mathcal{S}\rvert$
K_d	budget for the dynamic sensors
τ_0	time at which the epidemic is detected
τ_i, $i \in \mathbb{N}^+$	time at which the i^th dynamic sensor is placed
δ	placement delay, $\tau_i - \tau_{i-1} = \delta\ \forall i \in \mathbb{N}^+$
$\mathcal{D}_i$, $i \in \mathbb{N}^+$	set of dynamic sensors at time τ_i
$\mathcal{O}_i$, $i \in \mathbb{N}$	set of observations at time τ_i
$\omega = (u_\omega, t_\omega)$	observation of node u_ω: if u_ω is infected, t_ω is its infection time if u_ω is not infected, $t_\omega = \emptyset$
$\mathcal{B}_i$, $i \in \mathbb{N}$	set of candidate sources given $\mathcal{O}_i$
$\mathcal{C}_i$, $i \in \mathbb{N}^+$	set of candidate dynamic sensors at τ_i

we have
$$\mathcal{O}_i \backslash \mathcal{O}_{i-1} = \Big\{(d_i, t_{d_i})\Big\} \bigcup \Big\{(u, t_u) : u \in (\mathcal{S} \cup \mathcal{D}_{i-1}) \cap \mathcal{I}_i\Big\}.$$

Candidate Dynamic Sensors. The set of nodes among which we can choose a dynamic sensor at time τ_i is called $\mathcal{C}_i$. Clearly, $\mathcal{C}_1 = V \backslash \mathcal{S}$ and, for $i \geq 2$, $\mathcal{C}_i = V \backslash (\mathcal{S} \cup \mathcal{D}_{i-1})$.

Candidate Sources. At step i, v is a *candidate source* if, conditioned on $\mathcal{O}_i$ it has a non-zero probability of being the source. $\mathcal{B}_i$ is the set of candidate sources at step i, i.e.,

$$\mathcal{B}_i \triangleq \{v \in V : \mathrm{P}(v = v^\star | \mathcal{O}_i) > 0\}. \quad (1)$$

In particular, the initial set of candidate sources is

$$\mathcal{B}_0 = \{v \in V : \mathrm{P}(v = v^\star | \mathcal{O}_0) > 0\}.$$

Double Metric Dimension. Finally we recall the definition of Double Resolving Set (DRS) and Double Metric Dimension (DMD) of a network [3], which will be useful in the following sections.

Given a network $\mathcal{G}(V,E)$, a DRS is a subset $\mathcal{Z} \subseteq V$ such that for every $v_1, v_2 \in V$ there exist $z_1, z_2 \in \mathcal{Z}$ such that $d(v_1,z_1) - d(v_2,z_1) \neq d(v_1,z_2) - d(v_2,z_2)$, i.e., v_1, v_2 can be *distinguished* based on their distances to z_1, z_2. We will use the following lemma [6].

LEMMA 1. *Let $\mathcal{Z}$ be a DRS containing z'. Then, for every $v_1, v_2 \in V$ there exists $z'' \in \mathcal{Z}$ such that $d(v_1,z') - d(v_2,z') \neq d(v_1,z'') - d(v_2,z'')$.*

When an epidemic spreads on $\mathcal{G}$ and the transmission delays are deterministic, the infection times of a DRS suffice for distinguishing between any two possible sources [6]. The minimum size of a DRS of $\mathcal{G}$ is called the DMD of $\mathcal{G}$. Computing the DMD of a network is NP-hard [6]. Finding the set $\mathcal{U}$ of k nodes that maximize the number of nodes that are distinguished by any two nodes in $\mathcal{U}$ is also a NP-hard problem to which we refer as k-DRS [30]. An approximate solution of k-DRS can be found with a natural greedy heuristic [30] (see the extended version [31] for details). With a slight abuse of notation we denote by *k-DRS* the set $\mathcal{Z}$, such that $\lvert\mathcal{Z}\rvert = k$, obtained via the latter heuristic.

3. ONLINE SENSOR PLACEMENT & SOURCE LOCALIZATION

3.1 Deterministic Transmission Delays

For ease of exposition, we first present our algorithm in the case of deterministic transmission delays, i.e., $\theta_{uv} = w_{uv}$. In Section 3.2 we will show that our results naturally extend to random delays.

The following lemma formalizes that, when epidemics spread deterministically, the only source of randomness is the position of $v^\star$.

LEMMA 2. *Let $i \in \mathbb{N}^+$ and let $\mathcal{O}_i$ be the set of observations collected before or at τ_i. Then, $\mathrm{P}(\mathcal{O}_i | v = v^\star) \in \{0,1\}$.*

Since the starting time $t^\star$ of the epidemic is unknown, no single observation taken in isolation is informative about the position of the source (see Assumption (B.2)). Instead, two (or more) observations can become informative (which explains the importance of DMD and DRS for source localization). For this reason, we only consider the probability of two or more observations together. Let $\omega_1 \triangleq (u, t_u)$, and $\omega_2 \triangleq (w, t_w)$ two observations. If $t_u, t_w \neq \emptyset$, we define the event $\{\omega_1, \omega_2\}$ as $\{\omega_1, \omega_2\} \triangleq \{v = v^\star : d(v,u) - d(v,w) = t_u - t_w\}$. If $t_u \neq \emptyset$, $t_w = \emptyset$ and j is the smallest integer such that $\omega_2 \in \mathcal{O}_j$ for $j \in \mathbb{N}^+$, i.e., $\omega_2 \in \mathcal{O}_j \backslash \mathcal{O}_{j-1}$, we define $\{\omega_1, \omega_2\} \triangleq \{v = v^\star : d(v,u) - d(v,w) < t_u - \tau_j\}$.

We have the following lemma, which immediately follows from the definitions above.

LEMMA 3. *Let $\omega_1 \triangleq (u, t_u)$, and $\omega_2 \triangleq (w, t_w)$ be two observations, then*

(a) if $t_u, t_w \neq \emptyset$, then $\mathrm{P}(\{\omega_1, \omega_2\} | v = v^\star) = 1$ if and only if $d(v,u) - d(v,w) = t_u - t_w$.

(b) if $t_u \neq \emptyset$, $t_w = \emptyset$ and j is the smallest integer such that $\omega_2 \in \mathcal{O}_j$ for $j \in \mathbb{N}^+$, then $\mathrm{P}(\{\omega_1, \omega_2\} | v = v^\star) = 1$ if and only if $d(v,u) - d(v,w) < t_u - \tau_j$.

Algorithm description. The key idea is to iteratively choose the most informative node as a dynamic sensor. At every step i, we first select as new dynamic sensor d_i the node that maximizes the expected improvement (*gain*) in the localization process; then, we compute $\mathcal{B}_i$ using the information given by the dynamic sensor d_i and by the nodes in $\mathcal{S} \cup \mathcal{D}_{i-1}$ that became infected in $(\tau_{i-1}, \tau_i]$. The pseudocode for our algorithm is given in Algorithm 1.

The running time of Algorithm 1 depends on the definition of GAIN and will be discussed at the end of this section. We describe the functions INITIALIZECANDSOURCES, UPDATE and GAIN in the following subsections.

Initial Candidate-Sources Set $\mathcal{B}_0$. Based on the first observation available (i.e., the infection time τ_0 of the first infected static sensors $\mathcal{S}_0 \subseteq \mathcal{S}$), the initial set of candidate sources $\mathcal{B}_0$ contains all nodes that are closer to $\mathcal{S}_0$ than to $\mathcal{S} \backslash \mathcal{S}_0$.

PROPOSITION 1. *Let $\mathcal{S}_0$ be the set of the first infected static sensors and $\mathcal{O}_0$ be the first observation set. For every $v \in V$, let $\mathcal{S}_0^v$ be the set of the static sensors that are at minimum distance from v, i.e., $\mathcal{S}_0^v = \{s \in \mathcal{S} : d(v,s) = \min_{r \in \mathcal{S}} d(v,r)\}$. Then, $v \in \mathcal{B}_0$ if and only if $\pi(v) > 0$ and $\mathcal{S}_0^v = \mathcal{S}_0$.*

Algorithm 1 Online Sensor Placement & Source Localization

Require: K_d *budget for dynamic sensors*
Require: Set $\mathcal{S}$ of *static sensors*, set $\mathcal{O}_0$ of *initial observations*
  budget $\leftarrow K_d$
  $\mathcal{B}_0 \leftarrow$ InitializeCandSources($\mathcal{S}$, $\mathcal{O}_0$) *cand. sources*
  $\mathcal{C}_1 \leftarrow V \backslash \mathcal{S}$ *candidate-sensors*
  $\mathcal{D}_0 \leftarrow \{\}$, time $\leftarrow \tau_0 + \delta$, $i \leftarrow 1$
  while $|\mathcal{B}_{i-1}| > 1$ and budget > 0 **do**
    $d_i \leftarrow \arg\max_{c \in \mathcal{C}_i}$ Gain$(c, \mathcal{B}_{i-1})$
    $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1} \cup \{d_i\}$
    $\mathcal{O}_{i+1} \leftarrow \mathcal{O}_i \cup \{$new observations$\}$
    $\mathcal{B}_i \leftarrow$ Update$(\mathcal{B}_{i-1}, \mathcal{O}_i)$
    $\mathcal{C}_{i+1} \leftarrow \mathcal{C}_i \backslash d_i$
    time $\leftarrow$ time $+\delta$, budget $\leftarrow$ budget -1, $i \leftarrow i+1$
  end while
  return $\mathcal{B}_{i-1}$

Proof. By definition of $\mathcal{B}_0$, $v \in \mathcal{B}_0$ if and only if $\mathrm{P}(v = v^\star|\mathcal{O}_0) > 0$. In the deterministic setting any $\mathcal{O}_0$ collected from a given epidemic has non-zero probability, hence $\mathrm{P}(\mathcal{O}_0) > 0$. Now,

$$\mathrm{P}(v = v^\star|\mathcal{O}_0) = \mathrm{P}(\mathcal{O}_0|v = v^\star)\pi(v)/\mathrm{P}(\mathcal{O}_0) > 0$$

if and only if $\pi(v) > 0$ and $\mathrm{P}(\mathcal{O}_0|v = v^\star) > 0$. Hence, by Lemma 2, $\mathrm{P}(\mathcal{O}_0|v = v^\star) = 1$, which means that v is at distance $\min_{r \in \mathcal{S}} d(v, r)$ from all static sensors in $\mathcal{S}_0$ and at distance larger than $\min_{r \in \mathcal{S}} d(v, r)$ from all nodes in $\mathcal{S}\backslash\mathcal{S}_0$, i.e., $\mathcal{S}_0^v = \mathcal{S}_0$. □

Update. We now show how the set of candidate sources is updated at every step.

Lemma 4. *Let $i \in \mathbb{N}^+$. Then, $\mathcal{B}_i \subseteq \mathcal{B}_{i-1}$.*

Proof. Let $v \in \mathcal{B}_{i-1}$. Since $\mathcal{O}_{i-1} \subseteq \mathcal{O}_i$, $\mathrm{P}(v = v^\star|\mathcal{O}_i) > 0$ implies $\mathrm{P}(v = v^\star|\mathcal{O}_{i-1}) > 0$ and, from (1), we have that $\mathcal{B}_i \subseteq \mathcal{B}_{i-1}$. □

Using Lemma 4, at step i, we compute the set of candidate sources $\mathcal{B}_i$ based on $\mathcal{B}_{i-1}$ and on $\mathcal{O}_i\backslash\mathcal{O}_{i-1}$. More specifically, in Update we compute $\mathcal{B}_i$ by applying Proposition 2.

Proposition 2. *Let $i \in \mathbb{N}^+$ and take $s_0 \in \mathcal{S}_0$ arbitrarily. Moreover, for $\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}$, define the set $\mathcal{B}_\omega^i$ as*

$$\mathcal{B}_\omega^i \triangleq \begin{cases} \{v \in \mathcal{B}_{i-1} : d(u_\omega, v) - d(s_0, v) = t_\omega - \tau_0\}, & \text{if } t_\omega \neq \emptyset \\ \{v \in \mathcal{B}_{i-1} : d(u_\omega, v) - d(s_0, v) > \tau_i - \tau_0\}, & \text{if } t_\omega = \emptyset. \end{cases} \tag{2}$$

Then, $\mathcal{B}_i = \bigcap_{\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}} \mathcal{B}_\omega^i$.

Proof. The proof is decomposed in the following steps:

(A) $\mathcal{O}_i\backslash\mathcal{O}_{i-1} = \{\omega\}$, $t_\omega \neq \emptyset \Rightarrow \mathcal{B}_i = \mathcal{B}_\omega^i$

(B) $\mathcal{O}_i\backslash\mathcal{O}_{i-1} = \{\omega\}$, $t_\omega = \emptyset \Rightarrow \mathcal{B}_i = \mathcal{B}_\omega^i$

(C) $\mathcal{B}_i = \bigcap_{\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}} \mathcal{B}_\omega^i$.

(A) Let $\mathcal{O}_i\backslash\mathcal{O}_{i-1} = \{\omega\}$ and $t_\omega \neq \emptyset$.
(i) We show first that $\mathcal{B}_i \subseteq \mathcal{B}_\omega^i$. Let $\omega_0 \triangleq (s_0, \tau_0) \in \mathcal{O}_0$ and take $v \in \mathcal{B}_i$. Because of (1), $\mathrm{P}(v = v^\star|\mathcal{O}_i) > 0$. This implies that $\mathrm{P}(v = v^\star|\{\omega_0, \omega\}) > 0$. Applying Lemma 4 recursively, we have that $v \in \mathcal{B}_0$ and therefore $\pi(v) > 0$ because of Prop. 1. With $\mathrm{P}(v = v^\star|\{\omega_0, \omega\}) > 0$, this implies that $\mathrm{P}(\{\omega_0, \omega\}|v = v^\star) > 0$. By Lemma 2, we have that $\mathrm{P}(\{\omega_0, \omega\}|v = v^\star) = 1$. Hence v satisfies $d(u_\omega, v) - d(s_0, v) = t_\omega - \tau_0$ and $v \in \mathcal{B}_\omega^i$.
(ii) We show that $\mathcal{B}_\omega^i \subseteq \mathcal{B}_i$. Let $v \in \mathcal{B}_\omega^i$. In order to show that $\mathrm{P}(v = v^\star|\mathcal{O}_i) > 0$, it suffices to show that for any two observations $\omega_1, \omega_2 \in \mathcal{O}_i$, $\mathrm{P}(\{\omega_1, \omega_2\}|v = v^\star) = 1$, since then we also have that $\mathrm{P}(\mathcal{O}_i|v = v^\star) = 1$, which implies in turn that $\mathrm{P}(v = v^\star|\mathcal{O}_i) > 0$ with a similar Bayesian argument as in the proof of Prop. 1. Therefore, we only have to prove that $\mathrm{P}(\{\omega_1, \omega_2\}|v = v^\star) = 1$ for any $\omega_1, \omega_2 \in \mathcal{O}_i$. If $\omega_1, \omega_2 \in \mathcal{O}_{i-1}$, since $v \in \mathcal{B}_{i-1}$ because of (2), $\mathrm{P}(v = v^\star|\{\omega_1, \omega_2\}) > 0$, hence, as in (A)(i), $\mathrm{P}(\{\omega_1, \omega_2\}|v = v^\star) = 1$. Let us assume, without loss of generality that $\omega_1 \triangleq (z, t_z) \in \mathcal{O}_{i-1}$ and $\omega_2 \triangleq \omega = (u_\omega, t_\omega)$. Then (2) implies that

$$d(u_\omega, v) - d(s_0, v) = t_\omega - \tau_0, \tag{3}$$

and two situations can arise depending on t_z.
a) $t_z \neq \emptyset$. Since $v \in \mathcal{B}_{i-1}$ and $\omega_1 \in \mathcal{O}_{i-1}$, by Lemmas 2 and 3, $d(z, v) - d(s_0, v) = t_z - \tau_0$. Together with (3), this implies that $d(u_\omega, v) - d(z, v) = t_\omega - t_z$ and, by Lemma 3 we conclude that $\mathrm{P}(\{\omega_1, \omega_2\}|v = v^\star) = 1$.
b) $t_z = \emptyset$. Let $j \in \mathbb{N}$ be the smallest integer such that $\omega_1 \in \mathcal{O}_j$. Since $v \in \mathcal{B}_{i-1}$ and $\omega_1 \in \mathcal{O}_{i-1}$ we have by Lemmas 2 and 3 that $d(z, v) - d(s_0, v) > \tau_j - \tau_0$. Together with (3), this implies $d(z, v) - d(u_\omega, v) > \tau_j - t_\omega$ and, by Lemma 3, we conclude that $\mathrm{P}(\{\omega_1, \omega_2\}|v = v^\star) = 1$.

(B) The proof follows similarly to (A).

(C) If $v \in \mathcal{B}_\omega^i$ for all $\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}$, by (2), we have that $\mathrm{P}(\{\omega, \omega_0\}|v = v^\star) = 1$ for all $\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}$. By a reasoning similar to (A)(ii), this implies that $\mathrm{P}(\mathcal{O}_i|v = v^\star) = 1$, hence $v \in \mathcal{B}_i$ and $\bigcap_{\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}} \mathcal{B}_\omega^i \subseteq \mathcal{B}_i$. Moreover, if $v \notin \bigcap_{\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}} \mathcal{B}_\omega^i$, then $\mathrm{P}(\{\omega, \omega_0\}|v = v^\star) = 0$ for some $\omega \in \mathcal{O}_i\backslash\mathcal{O}_{i-1}$, hence $v \notin \mathcal{B}_i$. □

Correctness of Algorithm 1. We are now ready to prove the correctness of Algorithm 1, which, in fact, does not depend on the definition of Gain: As we will see in Section 4, Gain has an effect on the convergence speed of Algorithm 1 but not on the localization of the source.

Theorem 1. *Let the budget for the dynamic sensors be unrestricted ($K_d = \infty$). Algorithm 1 always returns $\{v^\star\}$.*

Proof. From Prop. 1, it follows that $v^\star \in \mathcal{B}_0$. Moreover, from Prop. 2, it follows that $v^\star \in \mathcal{B}_i$ at every step i of the algorithm. Thus, it only remains to prove that we make progress, i.e., that for any $v \in \mathcal{B}_0\backslash\{v^\star\}$, there is a step i such that $v \notin \mathcal{B}_i$. By Lemma 1, for any $v \in \mathcal{B}_0\backslash\{v^\star\}$ and $s_0 \in \mathcal{S}_0$, there exists $w \in V$ such that $d(v, w) - d(v^\star, w) \neq d(v, s_0) - d(v^\star, s_0)$. Let $i \in \mathbb{N}^+$ be the first step such that the infection time t_w of w satisfies $t_w \leq \tau_i$. Then, if $w \in \mathcal{S}$, we have $v \notin \mathcal{B}_{(w,t_w)}^i$ (where $\mathcal{B}_{(w,t_w)}^i$ is defined by (2)) and hence $v \notin \mathcal{B}_i$. If $w \notin \mathcal{S}$, let $j \in \mathbb{N}^+$ be the iteration step at which we choose w as a sensor. Then, for $\ell = \max(i, j)$, $v \notin \mathcal{B}_{(w,t_w)}^\ell$, and hence $v \notin \mathcal{B}_\ell$. □

We know from Prop. 2 that every new observation potentially reduces the number of candidate sources and makes

the localization progress. At each step of Algorithm 1, GAIN evaluates the expected progress in localization for all candidate sensors and we choose as dynamic sensor the node that yields to the maximum value. We consider three possible GAIN functions: SIZE-GAIN, DRS-GAIN and RC-GAIN. It is not *a priori* clear which version of GAIN leads to a faster convergence. Hence, we experiment with all of them.

SIZE-GAIN. Perhaps the most natural GAIN function is the one that computes the expected reduction in the number of candidate sources. Call $\mathcal{B}_i^{(c)}$ the set of candidate sources after adding c as dynamic sensor at step i. We define the SIZE-GAIN of adding c at step i as $g_i^{\text{SIZE}}(c) \triangleq \mathbf{E}[|\mathcal{B}_{i-1}| - |\mathcal{B}_i^{(c)}|]$. In practice, $g_i^{\text{SIZE}}(c)$ can be easily computed by summing over the set $\mathcal{T}_i^c$ of the possible infection times for c (see Definition 1).

DEFINITION 1. *Let $i \in \mathbb{N}^+$ and $\mathcal{C}_i$ be the set of possible dynamic sensors at step i. Let $c \in \mathcal{C}_i$. Then,*

$$\mathcal{T}_i^c \triangleq \{h \in (-\infty, \tau_i] : h = d(v,c) - d(v,s_0) - \tau_0 \text{ for some } v \in \mathcal{B}_{i-1}\} \quad (4)$$

is the set of possible infection times of c by step i.

PROPOSITION 3. *Let $i \in \mathbb{N}^+$ and $\mathcal{C}_i$ be the set of possible dynamic sensors at step i. Let $c \in \mathcal{C}_i$. For $h \in \mathcal{T}_c$, define*

$$\begin{aligned} b_i(c,h) &\triangleq \{v \in \mathcal{B}_{i-1} : \mathrm{P}(v = v^\star | t_c = h) > 0\} \\ &= \{v \in \mathcal{B}_{i-1} : h = d(v,c) - d(v,s_0) + \tau_0\}, \\ \tilde{b}_i(c) &\triangleq \{v \in \mathcal{B}_{i-1} : \mathrm{P}(v = v^\star | t_c > \tau_i) > 0\} \\ &= \{v \in \mathcal{B}_{i-1} : \tau_i < d(v,c) - d(v,s_0) + \tau_0\}. \end{aligned}$$

$$\text{Then, } g_i^{\text{SIZE}}(c) = \sum_{h \in \mathcal{T}_c} \pi(b_i(c,h)) \cdot (|\mathcal{B}_{i-1}| - |b_i(c,h)|) + \pi(\tilde{b}_i(c)) \cdot (|\mathcal{B}_{i-1}| - |\tilde{b}_i(c)|). \quad (5)$$

DRS-GAIN. The definition of this GAIN is inspired by the notion of DRS (see Section 2). After the first static sensor is infected, it is clearly possible to detect the source with at most DMD($\mathcal{B}_0$) additional observations. Indeed, observing the infection times of a DRS of $\mathcal{B}_0$ removes all ambiguities about the source identity. DRS-GAIN is a *dynamic* greedy implementation of this observation, where at each step i we choose the sensor that gives the most progress in forming a DRS of $\mathcal{B}_i$. Let $c \in \mathcal{C}_i$ and let $X_c = 1$ if there exists $v \in \mathcal{B}_{i-1}$ such that the infection time t_c of c is larger than τ_i (i.e., such that $d(v,c) - d(v,s_0) - \tau_0 > \tau_i$), $X_c = 0$ otherwise. Then, the value of DRS-GAIN at step i is

$$g_i^{\text{DRS}}(c) \triangleq |\mathcal{T}_i^c| + X_c. \quad (6)$$

Note that both SIZE-GAIN and DRS-GAIN account only for the benefit of adding the dynamic sensor c: For tractability, we ignore all observations $\omega \in \mathcal{O}_i \backslash \mathcal{O}_{i-1}$ such that $u_\omega \neq c$.

RC-GAIN. RC-GAIN (*Random-Candidate*-GAIN) assigns gain 1 to all candidates sources and gain 0 to all nodes that are not candidate sources: At step i, for $c \in \mathcal{C}_i$ we set $g^{\text{RC}}(c) = 1$ if $c \in \mathcal{B}_{i-1}$, $g^{\text{RC}}(c) = 0$ otherwise. In other words, we randomly choose the dynamic sensors among the candidate sources. Note that if the infection time of at least one node in $\mathcal{B}_{i-1}$ is already observed, adding a sensor in any other node in $\mathcal{B}_{i-1}$ implies $|\mathcal{B}_i| \leq |\mathcal{B}_{i-1}|$. Hence, this very simple GAIN ensure that the source-localization makes progress at each step.

Running time. In the worst case, the **while** loop of Algorithm 1 is entered N times. At step i, both the UPDATE and the computation of any of the proposed GAIN functions takes $O(|\mathcal{B}_i|)$ steps. Hence, with the proposed definitions of GAIN, the i^{th} iteration takes $O(|\mathcal{C}_i| \cdot |\mathcal{B}_i|) \subseteq O(N^2)$. Although the running time can potentially reach $\Theta(N^3)$, our experiments show that, in many practical cases, $|\mathcal{B}_i|$ is sublinear.

3.2 Non-Deterministic Transmission Delays

In this section we assume that the transmission delays are independent continuous random variables such that, for every $uv \in E$, the support of the transmission delay θ_{uv} is bounded and symmetric with respect to w_{uv}, i.e., is $[w_{uv}(1-\varepsilon), w_{uv}(1+\varepsilon)]$, with $\varepsilon \in [0,1]$. We refer to ε as *noise parameter*. For $\varepsilon > 0$, the transmission delay over an edge of weight w can deviate up to εw from its expected value. $\varepsilon = 0$ corresponds to the deterministic model of Section 3.1.

The structure of the algorithm for sensor placement and source localization is identical to that of Algorithm 1, the only changes are in INITIALIZECANDSOURCES and UPDATE.

The following proposition characterizes the candidate sources at step i through necessary conditions. It is used in INITIALIZECANDSOURCES and in UPDATE to discard, at step i, the nodes v such that $\mathrm{P}(v = v^\star | \mathcal{O}_i) = 0$.

PROPOSITION 4. *Let s_0 be the first infected sensor, that is infected at time τ_0 and let $i \in \mathbb{N}^+$.*

1. *If $v \in \mathcal{B}_0$, then*

$$d(s_0,v) - \min_{s \in \mathcal{S}} d(v,s) \leq \varepsilon(d(s_0,v) + \min_{s \in \mathcal{S}} d(v,s)).$$

2. *Let $\omega_1, \omega_2 \in \mathcal{O}_i$ with $t_{\omega_i} \neq \emptyset$ for $i \in \{1,2\}$. If $v \in \mathcal{B}_i$, then*

$$|d(u_{\omega_2},v) - d(u_{\omega_1},v) - t_{\omega_2} + t_{\omega_1}| \leq \varepsilon(d(u_{\omega_1},v) + d(u_{\omega_2},v)). \quad (7)$$

3. *Let $\omega_1, \omega_2 \in \mathcal{O}_i$ with $t_{\omega_1} \neq \emptyset$, $t_{\omega_2} = \emptyset$ and let $\omega_2 \in \mathcal{O}_i$. If $v \in \mathcal{B}_i$, then*

$$\tau_i - t_{\omega_1} - d(u_{\omega_2},v) + d(u_{\omega_1},v) < \varepsilon(d(u_{\omega_1},v) + d(u_{\omega_2},v)). \quad (8)$$

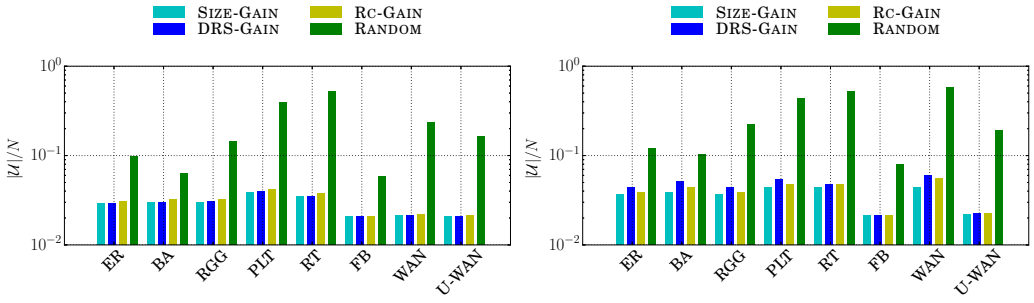
PROOF. Follows from $\theta_{uv} \in [w_{uv}(1-\varepsilon), w_{uv}(1+\varepsilon)]$ for every $uv \in E$. □

Prop. 4 is similar in spirit to Prop. 2. Note in particular, that by setting $\varepsilon = 0$ in (7) and (8) we get, for two arbitrary observations $\omega_1, \omega_2 \in \mathcal{O}_i$, the respective of the conditions on the infection times used to define $\mathcal{B}_\omega^i$ in (2). However, differently from Prop. 2, when $\varepsilon > 0$, we cannot give necessary and sufficient conditions for a node to be the source by simply comparing all observations with a reference observation. Hence, when $\varepsilon > 0$, at step i the function UPDATE keeps in $\mathcal{B}_i$ only the nodes such that both (7) and (8) hold for any $\omega_1, \omega_2 \in \mathcal{O}_i$. This increases the running time of iteration i by at most $O(\mathcal{S} \cup \mathcal{D}_i)$.

Correctness of Algorithm 1. The correctness of Theorem 1 also holds when the transmission delays are non-deterministic and is independent of the definition of GAIN.

	ER (p=0.016)	BA (m=2)	RGG (R=0.3)	RT	PLT	FB	U-WAN	WAN
$\|V\|$	250	250	250	250	250	3732	2258	2258
$\|E\|$	511	496	696	249	249	82305	17695	17695
avg degree	4.09	3.96	5.6	1.99	1.99	44.1	15.67	15.67
avg shortest path	4.09	3.47	9.68	7.45	37.8	5.34	6.94	3.56
avg clustering	0.02	0.06	0.56	0	0	0.54	0.65	0.65

Table 1: Statistics for the networks considered in the experiments.



(a) $|\mathcal{U}|/N$ ($\varepsilon = 0$) (b) $|\mathcal{U}|/N$ ($\varepsilon = 0.2$)

Figure 1: Relative cost of source localization.

THEOREM 2. *Let $\varepsilon \in [0,1]$ and θ_{uv} be a continuous random variable with support $[(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}]$ for every $uv \in E$. Moreover let the budget for dynamic sensors be unrestricted ($K_d = \infty$). Algorithm 1 always returns $\{v^\star\}$.*

PROOF. The proof follows the structure of that of Theorem 1. First note that nodes are removed from the set of candidate sources if and only if they do not satisfy some of the necessary conditions expressed by inequalities (7) and (8). Hence, because of Proposition 4, the source $v^\star$ is never removed from the set of candidates. Next, we want to prove that, for every node $v \neq v^\star$, there exists a node $w \in V$ such that, when the infection time of w is observed, v is removed from the set of candidate sources. Take $w = v^\star$ and suppose that its infection time t_w is observed. Let $v \neq w$ be another node for which the infection t_v time is also observed. As $w = v^\star$, we have $t_v > t_w$. Note that inequality (7) cannot hold for v and w: Indeed, we would have $0 < (1-\varepsilon)d(v,w) \leq t_w - t_v < 0$, which gives a contradiction. Let $i \in \mathbb{N}^+$ such that $w, v \in \mathcal{S} \cup \mathcal{D}_i$ and such that t_v is smaller than τ_i. Then, $v \notin \mathcal{B}_i$. □

GAIN. Building on the deterministic case, we can compute an approximate version of the SIZE-GAIN value $g_i^{SIZE}(c)$ for the case in which $\varepsilon \neq 0$. For the details of this computation see the extended version [31]. DRS-GAIN and RC-GAIN do not depend on the epidemic model, hence remain unchanged with respect to Section 3.1.

Approximate Source Localization. When $K_d < \infty$ and the convergence of the algorithm is not guaranteed, we could consider substituting ε with $\tilde{\varepsilon} = C\varepsilon$, $0 < C \leq 1$, in inequalities (7) and (8). Here, C plays the role of a tolerance constant. Intuitively, when C is small, we quickly narrow the candidate sources set, but the probability that the correct source is not identified by the algorithm is high; when C is large, the probability that the algorithm identifies the real source as a candidate source is high, but possibly we have many false positives. The setting $C < 1$ can be interesting for the case in which the transmission delays θ_{uv} are not uniform, e.g., when the delays are more concentrated around their expected value values. A study of this extension is left for future work.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

In our experiments, the *transmission delays* are *uniformly distributed*. The uniform distribution is, among the unimodal distributions on a bounded support, the one that maximizes the variance [13]. Hence, uniform delays are a very challenging setting for source localization.

The choice of static sensors is inspired by the work of Spinelli et al. [30], where static sensor placement is extensively studied. We let $\mathcal{S} = k\text{-}DRS$ with $k = K_s$ (see Section 2), so that the number of nodes that are *distinguished* by the static sensors is maximized.[1] We also do not evaluate the impact of the *budget* K_s, rather we are concerned with decreasing total number of sensors $|\mathcal{U}|$. We set $K_s = 0.02 \cdot N$.

A study of different static placement strategies and of the trade-off between K_s and the timeliness of source localization is left for future work.

We evaluate the performance of the different approaches in terms of the *(relative) cost* of the sensor placement, i.e., the fraction $|\mathcal{U}|/N$ of the sensors used for localization. All results are averaged over at least 100 simulations in which the position of the source is chosen uniformly at random.

The *placement delay* δ, unless otherwise specified, is $\delta = 1$. This means that the epidemic and the localization process have approximately the same speed, which we believe is a

[1]The optimal choice of the static sensors depends on the objective considered. For example, an alternative goal might be to minimize the expected time before the first static sensor is infected, for which one would choose a K_s-Median [16] set as $\mathcal{S}$.

realistic assumption in many applications. Moreover, in Section 4.3 we present an experiment that evaluates the effect of this parameter and in which $\delta = 1$ emerges as a good trade-off between the cost of the algorithm and the time needed for detection (see Figure 3).

Algorithms & Baselines. We study the performance of Algorithm 1 for Size-Gain, Drs-Gain and RC-Gain (see Section 3.1).

As recalled in Section 2, with a static sensor placement (i.e, $K_d = 0$), the minimum number of sensors required to localize the source when the transmission delays are deterministic is the DMD of the network [6]. Hence, we use DMD as one natural benchmark for the cost of our algorithm.

Moreover we compare with the following baselines:

- ◇ Random. We run Algorithm 1 but, at each step i, we select d_i at random from $V \setminus (\mathcal{S} \cup \mathcal{D}_{i-1})$.
- ◇ AllStatic. When $K_d < N$, we compare the performance of Algorithm 1 (with K_s static and K_d dynamic sensors) with an entirely static version of Algorithm 1 where the budget for static sensors is $K'_s = K_s + K_d$ and the budget for dynamic sensors is $K'_d = 0$.

4.2 Network Topologies

We consider both synthetic and real-world networks; the network properties and statistics are reported in Table 1.

Synthetic networks. We generated synthetic networks from the following classes: Erdös-Rényi networks (ER) [10], Barabási-Albert networks (BA) [2], Random Geometric Graph on the sphere (RGG) [25], regular trees of degree 3 (RT) and trees with power-law distributed node degree (PLT). For each network class, 10 connected instances of size 250 with unit edge weights were generated.

Real-world networks. *Facebook Egonets (FB).* This dataset is a subset of the Facebook network, consisting of 3732 nodes. It was obtained from the union of 10 Facebook egonet networks [23] after removing the ego nodes[2] and taking the largest connected component. We set all weights to $w = 1$ as there is not a straightforward method for deriving realistic edge weights for this network.

World Airline Network (WAN). This network is obtained from a publicly available dataset [24] that provides the aircraft type used for every daily connection between over three thousands airports. Using this data we can derive the number of seats available on each route daily. We preprocess the network by removing the connections on which less than 20 seats per day are available and by assigning to each connection (u, v) the average between the number of seats available from u to v and from v to u. Also, we iteratively remove leaf nodes (for which we believe connections are not well represented in the dataset), and we obtain a network of 2258 nodes. The definition of the edge weights is inspired by a work by Colizza et al [7]. An edge (u, v) is weighted with an integer[3] approximation of the expected time between the infection of city u and the arrival of an infected individual at city j (see the extended version [31] for details). This gives a very skewed weight distribution. Our experiments show that the diversity of the edge weights brings an additional challenge to source localization. In order to evaluate the impact of non-uniform weights, we also run our experiments on an unweighted version (U-WAN) of this network (in which all weights are set to 1).

[2]The ego nodes were removed in order to ensure that the sampling of contacts across the nodes in the network is uniform.

[3]Integer weights actually make the problem *more difficult* when $\varepsilon = 0$ (because it is more difficult to distinguish among nodes based on their distances to the sensors); when $\varepsilon > 0$ the problem is again harder because we consider a continuous distribution for the transmission delays.

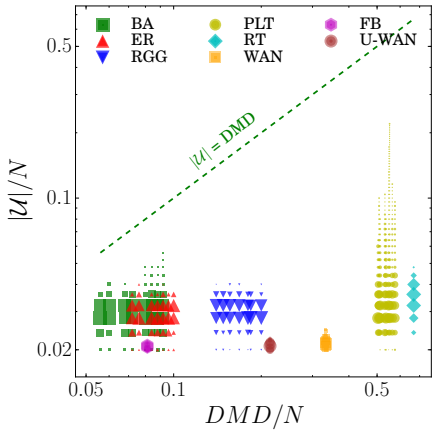


Figure 2: Sensors needed for source-localization by Algorithm 1 with Size-Gain and $\varepsilon = 0$ compared with the number needed by an optimal offline placement (DMD). Larger markers represent higher concentrations of data points.

4.3 Results

Different Gain functions. We study the effect of Gain on the performance of Algorithm 1. For each variant, i.e., Size-Gain, DRS-Gain, RC-Gain, and for the Random heuristic, we report the relative cost. We let $K_d = \infty$; hence, by Theorems 1 and 2, Algorithm 1 always localizes the source. We consider both a deterministic setting ($\varepsilon = 0$) and a non-deterministic setting with $\varepsilon = 0.2$, which means that the transmission delays can deviate up to 20% from their average value. The results are depicted in Figure 1(a)-1(b). We observe that for the real networks and $\varepsilon = 0$ all proposed Gain have similar performance. For FB and U-WAN, this is true also when $\varepsilon > 0$. These are also the cases where our algorithm has the smallest cost, hence we conclude that, when source localization is less challenging, Gain does not have a strong impact. In all other cases, Size-Gain consistently gives the best performance. The improvement with respect to Drs-Gain is most noticeable when $\varepsilon > 0$; indeed, in this setting Drs-Gain is outperformed by the simple RC-Gain. We attribute this to the fact that, when there is high variance in the transmission delays, splitting the candidate sources into subsets of nodes which have different average infection times (see the definition of Drs-Gain in Eq. (6)), does not guarantee that we are able to distinguish them based on the observed infection times [30]. Instead, as mentioned in Section 3.1, RC-Gain enforces a continuous progress in shrinking the set of candidate sources. Since Size-Gain emerges as the best Gain among those we consider, we will use it in the remaining experiments (unless otherwise specified).

DMD vs. Cost of Algorithm 1. We now focus on the deterministic case ($\varepsilon = 0$) when $K_d = \infty$, and compare $|\mathcal{U}|/N$ with the (approximate) DMD. We recall (see Section 2) that the DMD is the size of the optimal offline sensor placement

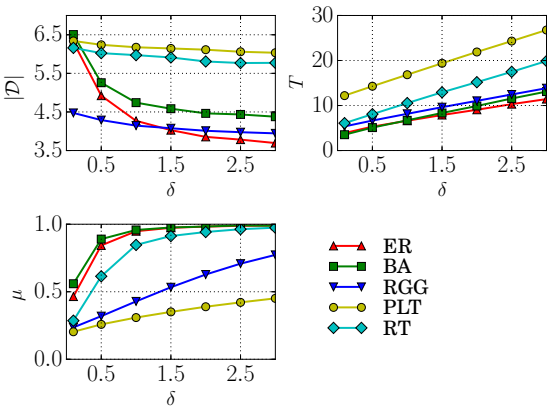
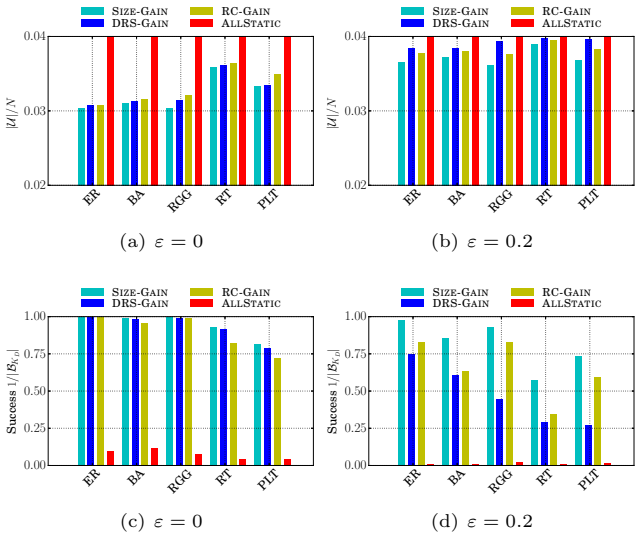


Figure 3: (Top left) Number $|\mathcal{D}|$ of dynamic sensors required to detect the source for different values of the placement delay δ; (Top right) Time T (in time units) until localization; (Bottom) Fraction μ of infected nodes at localization time. The noise parameter is $\varepsilon = 0.2$.



(a) $\varepsilon = 0$ (b) $\varepsilon = 0.2$

(c) $\varepsilon = 0$ (d) $\varepsilon = 0.2$

Figure 4: Average relative cost $|\mathcal{U}|/N$ and success $1/|\mathcal{B}_{K_d}|$ of source localization when $K_s = K_d = 0.02 \cdot N$.

for this setting. The results are depicted in Figure 2. For all topologies, $|\mathcal{U}|/N$ is much smaller than DMD$/N$. The improvement is particularly significant for trees where, on the one hand, DMD is very large (equal to the number of leaves [6]) and, on the other hand, the topology makes it easy for our algorithm to rapidly narrow the search for the source to a small set of candidates.

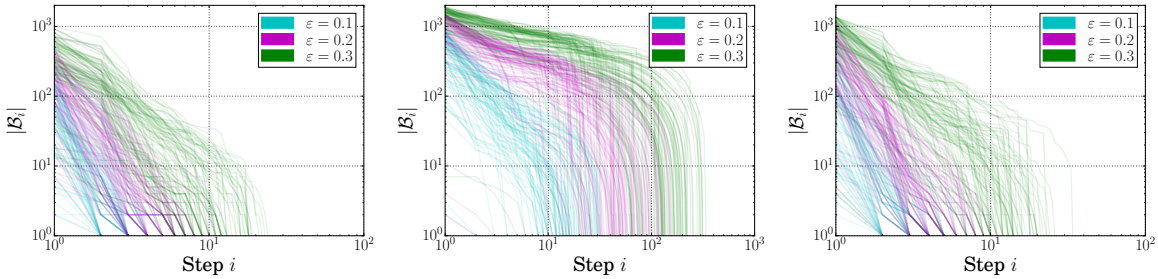
AllStatic vs. Algorithm 1. We look at the performance of Algorithm 1 when the budget for dynamic sensors is limited to a small fraction of nodes; we let $K_d = 0.02 \cdot N = K_d$.

We compare Algorithm 1 with different Gain (Size-Gain, DRS-Gain and RC-Gain) against the AllStatic baseline with $K'_d = 0$ and $K'_s = K_s + K_d = 0.04 \cdot N$ (see Section 4.1). As $K_d < \infty$, it is no longer guaranteed that we localize the source; instead we evaluate the *success* of an algorithm with the metric $1/|\mathcal{B}_{K_d}|$, where $\mathcal{B}_{K_d}$ is the set of candidate sources at the last iteration step. Hence, the success is 1 when the source is localized (since $|\mathcal{B}_{K_d}| = 1$), and is decreasing in the size of $\mathcal{B}_{K_d}$. Note that $|\mathcal{U}| \leq 0.04 \cdot N$ and, in particular, $|\mathcal{U}| < 0.04 \cdot N$, only if the source was localized with fewer than K_d dynamic sensors. The results are presented in Figure 4. We observe that our approach outperforms the static sensor placement in terms of the budget used by the algorithm. Furthermore, for both $\varepsilon = 0$ and $\varepsilon > 0$, our algorithm gives a much higher success in source localization than AllStatic. Among the Gain tested, Size-Gain is again the best one, giving both the higher success and the minimum cost.

Placement delay. An important parameter used by Algorithm 1 is the placement delay δ, i.e., the time between two consecutive placements of a dynamic sensor. On the one hand, the larger δ is, the smaller we expect the cost of our algorithm to be; on the other hand, the smaller δ is, the less time we expect to need for localizing the source, hence the fewer individuals are infected before we do so. We vary δ and look at the number $|\mathcal{D}|$ of dynamic sensors used, the fraction μ of infected individuals at the time of localization, and the time T between the beginning of the epidemic and the localization of the source[4] (see Figure 3). We observe a trade-off between $|\mathcal{D}|$ and both T and μ.

Cost of localization and size of $|\mathcal{B}_i|$ for real networks. Finally, we evaluate the cost of localization in the practical setting of real networks with random delays. Moreover, to estimate how the running time varies for different values of the noise parameter and for the different topologies considered, we look at how the cardinality of the candidate set $\mathcal{B}_i$ defined by Eq. (1) decreases along the successive steps. We note beforehand that the approximate DMD is 303 (around $0.08 \cdot N$) for the FB network, 751 (around $0.3 \cdot N$) for WAN and 484 for U-WAN. Hence, source localization is more challenging on the WAN network. This is confirmed by the results shown in Figure 5. On the FB network, with noise parameter $\varepsilon = 0.3$, the correct localization of the source is achieved with a total cost $|\mathcal{U}| \approx 0.025 \cdot N$ of sensors. The average number of sensors needed is slightly larger for the U-WAN network ($|\mathcal{U}| \approx 0.03 \cdot N$). We attribute this effect to the presence of *bottleneck* edges, i.e., edges that appear on many different shortest paths and make it difficult to estimate the source based on its distance to the sensors. This effect becomes even stronger with the weighted version of the WAN network (where the total cost needed is around $|\mathcal{U}| \approx 0.085 \cdot N$). This last result highlights that the high variability among the edge-weights makes source localization substantially more difficult, especially for $\varepsilon > 0$ (see Figure 1 for a comparison of the cost between deterministic and non-deterministic delays). Given the high regime of the noise parameter we consider and the small percentage of sensors deployed, we conclude that our algorithm outperforms most other approaches to source-localization, which either need more sensors or tolerate smaller amounts of noise.

[4]To choose δ, one must consider also the scale of edge weights, here, for simplicity of exposition, we ignore this aspect and experiment only with unweighted networks.



(a) FB (b) WAN (c) U-WAN

Figure 5: Cardinality of the candidate sources set $\mathcal{B}_i$ at successive steps of the algorithm.

5. RELATED WORK

We briefly review some important contributions to source localization (see [15] for an in-depth discussion).

Complete observation. The first source-estimator was proposed by Shah and Zaman [28] in 2009. This work, and many others that followed, rely on what is often called a *complete observation* of the epidemic (see Assumption (B.1) in Section 1) [37, 27, 32]. In these models, the source is estimated by maximum likelihood estimation (MLE).

The results of [28] have been extended in many ways, e.g., to the case of multiple sources [21] or to obtain a *local* source estimator [8]. An alternate line of work that also uses Assumption (B.1), allows the observed states to be *noisy*, i.e., potentially inaccurate. For example, a model in which it is not possible to distinguish between susceptible and recovered nodes was studied by Zhu et al. [39].

Partial observation. Follow-up work considers a *partial observation* setting where a randomly-selected fraction of nodes reveal their state [18, 40, 22, 33]. These works do not assume that the infection times are known (see Assumption (A.2)), hence they need a large fraction of the nodes to be sensors (typically more than 30%) to localize the source.

Static sensor placement. Other works address the problem of strategically selecting sensor nodes *a-priori*, i.e., finding a *static* sensor placement. In the deterministic setting (see Assumption (B.5)) some works considered the problem of *minimizing* the budget required for detecting the source. This question is similar to the one we address, except that we allow random transmission delays and, most importantly, we propose an online solution. On trees, under (B.2) and (B.5), the minimization of the number of sensors has been studied [34]. Without (B.2) and (B.4), but with (B.5), approximation algorithms have been developed by Chen et al. [6].

Budgeted sensor placement. In a network of N nodes, the minimal budget required for source-localization can go up to $N-1$, in which case the result of Chen et al. is not practical. Hence, researchers have looked into a *budgeted* version of the problem, i.e., how to place sensors given that only a limited number of them is available. In this direction, "common sense" approaches, e.g., using high-degree vertices, or centrality measures were first evaluated [26, 20]. Later, the budgeted optimization problem was solved on trees [5] (B.4). Without (B.4), a heuristic approach, based on the definition of Double Resolving Set of a graph (see Section 2), has been shown to outperform all previous heuristics [30]. Due to budget restrictions, none of the works mentioned above can guarantee exact source localization.

Sequential sensor placement. Working under (B.5) and (B.2), Zejnilovic et al. [35], proposed an algorithm that sequentially places sensors in order to localize the source *after* the epidemic has spread through the entire network. Adopting very different techniques, we propose a solution that selects the sensors *while* the epidemic evolves, enhancing both cost- and time-efficiency. Moreover, our approach works without (B.5) and (B.2).

Transmission delays. Several models for how the epidemic spreads have been studied [17]. Discrete-time transmission delays were initially very common (see Assumption (B.5)) [22, 27, 1]. Then, to better approximate realistic settings, continuous-time transmission models with varying distributions for the transmission delays have been adopted; e.g., exponential [28, 21], Gaussian [26, 20, 19, 36] or truncated Gaussians [30]. We consider general continuous bounded-support distributions that are tractable but yet versatile.

Other related work. Two-stage resource allocation is also studied in the context of *robust optimization* where, to reach some objective, we allocate a-priori only a part of the resources and another part is deployed, at a higher cost, when more information is available [14]. Another related line of work in the Artificial Intelligence field is that of *active learning* which studies how one can, based on sparse data, adaptively take a sequence of decisions in order to optimize a given objective [12].

6. REFERENCES

[1] F. Altarelli, A. Braunstein, L. Dall'Asta, A. Lage-Castellanos, and R. Zecchina. Bayesian inference of epidemics on networks via belief propagation. *Physical review letters*, 112(11), 2014.

[2] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286, 1999.

[3] J. Cáceres, M. Hernando, M. Mora, I. Pelayo, M. Puertas, C. Seara, and D. Wood. On the metric dimension of cartesian products of graphs. *SIAM J. Discrete Mathematics*, 21(2), 2007.

[4] S. Cauchemez and N. Ferguson. Likelihood-based estimation of continuous-time epidemic models from time-series data: application to measles transmission in london. *Journal of the Royal Society Interface*, 5(25), 2008.

[5] L. Celis, F. Pavetić, B. Spinelli, and P. Thiran. Budgeted sensor placement for source localization on trees. In *LAGOS*, 2015.

[6] X. Chen, X. Hu, and C. Wang. Approximability of the minimum weighted doubly resolving set problem. In *COCOON*, 2014.

[7] V. Colizza, A. Barrat, M. Barthélemy, and A. Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. of the National Academy of Sciences of the USA*, 103(7), 2006.

[8] W. Dong, W. Zhang, and C. Tan. Rooting out the rumor culprit from suspects. In *IEEE ISIT*, 2013.

[9] B. Ehrenberg. How much is your personal data worth? `https://www.theguardian.com/news/datablog/2014/apr/22/how-much-is-personal-data-worth`, 2014.

[10] P. Erdös and A. Rényi. On random graphs. *I. Publ. Math. Debrecen*, 6, 1959.

[11] M. Farajtabar, M. Gomez-Rodriguez, M. Zamani, N. Du, H. Zha, and L. Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *AISTATS*, 2015.

[12] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42, 2011.

[13] H. Gray and P. Odell. On least favorable density functions. *SlAM Review*, 9, 1967.

[14] A. Gupta, V. Nagarajan, and R. Ravi. Thresholded covering algorithms for robust and max-min optimization. In *International Colloquium on Automata, Languages, and Programming*, 2010.

[15] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communication Survey Tutorials*, 2014.

[16] O. Kariv and S. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM journal of Applied Mathematics*, 37, 1979.

[17] M. Lelarge. Efficient control of epidemics over random networks. In *SIGMETRICS/Performance*, 2009.

[18] A. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Ph. Review E*, 90(1), 2014.

[19] A. Louni, A. Santhanakrishnan, and K. Subbalakshmi. Identification of source of rumors in social networks with incomplete information. *ASE SocialCom*, 2015.

[20] A. Louni and K. Subbalakshmi. A two-stage algorithm to estimate the source of information diffusion in social media networks. *IEEE INFOCOM Workshop on Dynamic Social Networks*, 2014.

[21] W. Luo and W. Tay. Identifying infection sources in large tree networks. In *IEEE SECON*, 2012.

[22] W. Luo, W. Tay, and M. Leng. How to identify an infection source with limited observations. *IEEE Journal of Sel. Topics in Signal Processing*, 8(4), 2014.

[23] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, 2012.

[24] OpenFlights. Route dataset. `http://openflights.org/data.html#route`, 2012.

[25] M. Penrose. *Random Geometric Graphs*. Oxford Studies in Probability, 2003.

[26] P. Pinto, P. Thiran, and M. Vetterli. Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109, 2012.

[27] B. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? *IEEE ICDM*, 2012.

[28] D. Shah and T. Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on information theory*, 57, 2011.

[29] Z. Somda, M. Meltzer, H. Perry, N. Messonnier, U. Abdulmumini, G. Mebrahtu, M. Sacko, K. Touré, S. O. Ki, T. Okorosobo, W. Alemu, and I. Sow. Cost analysis of an integrated disease surveillance and response system: case of burkina faso, eritrea, and mali. *Cost Effectiveness and Resource Allocation*, 7(1), 2009.

[30] B. Spinelli, L. Celis, and P. Thiran. Observer placement for source localization: The effect of budgets and transmission variance. In *Allerton Conf.*, 2016.

[31] B. Spinelli, L. Celis, and P. Thiran. Back to the source: An online approach for sensor placement and source localization. `https://arxiv.org/pdf/1702.01056v1.pdf`, 2017.

[32] S. Sundareisan, J. Vreeken, and B. Prakash. Hidden hazards: Finding missing nodes in large graph epidemics. In *SIAM SDM*, 2015.

[33] H. Wang, P. Zhang, L. Chen, H. Liu, and C. Zhang. Online diffusion source detection in social networks. In *IEEE Neural Networks (IJCNN)*, 2015.

[34] S. Zejnilovic, J. Gomes, and B. Sinopoli. Network observability and localization of the source of diffusion based on a subset of vertices. In *Allerton Conf.*, 2013.

[35] S. Zejnilović, J. Gomes, and B. Sinopoli. Sequential observer selection for source localization. In *IEEE GlobalSIP*, pages 1220–1224, 2015.

[36] X. Zhang, Y. Zhang, T. Lv, and Y. Yin. Identification of efficient observers for locating spreading source in complex networks. *Physica A: Statistical Mechanics and its Applications*, 442, 2016.

[37] L. Zheng and C. Tan. A probabilistic characterization of the rumor graph boundary in rumor source detection. In *IEEE DSP*, 2015.

[38] K. Zhu, Z. Chen, and L. Ying. Locating the contagion source in networks with partial timestamps. *Data Mining and Knowledge Discovery*, 2015.

[39] K. Zhu and L. Ying. Information source detection in the SIR model: A sample path based approach. In *IEEE ITA*, 2013.

[40] K. Zhu and L. Ying. A robust information source estimator with sparse observations. *Computational Social Networks*, 1(1), 2014.