Localizing the Source of an Epidemic Using Few Observations

THÈSE N^{O} 8391 (2018)

PRÉSENTÉE LE 23 MARS 2018
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 3
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Brunella Marta SPINELLI

acceptée sur proposition du jury:

Prof. P. Frossard, président du jury
Prof. P. Thiran, Dr L. E. Celis, directeurs de thèse
Dr M. Gomez-Rodriguez, rapporteur
Prof. B. Sinopoli, rapporteur
Prof. M. Salathé, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

To my family

Acknowledgements

I am sincerely grateful to my advisors, Prof. Patrick Thiran and Dr. L. Elisa Celis, for giving me the opportunity to write this thesis and for accompanying me all along the way. I was fortunate to have two advisors who are really passionate about their research and never hesitate to invest all their energies and thoughts. Thank you, Patrick, for your patient advice and, in particular, for insisting on the quality of the results and of their communication. Thank you, Elisa, for your dynamism and your pragmatic attitude with which you encouraged me to look for possible ways in problems that I would have already classified as impossible to solve.

I thank Dr. Manuel Gomez-Rodriguez, Prof. Marcel Salathé, Prof. Bruno Sinopoli and Prof. Pascal Frossard for accepting to be on my thesis committee and for reviewing my thesis. I am also grateful to Prof. Matthias Grossglauser for co-leading the lab with great enthusiasm and for his sensitive and precious advise at a few difficult turns of this journey. I also thank the lab's administrative staff, Patricia and Angela, for all the help and support they provided. Many thanks to Holly for her enthusiastic proof-reading and for being such a cheerful person.

I learned about a lot from my co-author Filip Pavetić to whom I am very grateful for sharing his knowledge and for his contagious enthusiasm. I also learned (and not only a little) from two Master students, Ahmad and Mladen, with whom it was really a pleasure and an enriching experience to work. I thank Dr. Philippe Glaziou for giving me the opportunity to join WHO as an intern in the summer of 2015 and for his guidance and support during this time.

During my years in LCA 3.5 I had the pleasure of meeting many brilliant students and fun characters: I thank Farid, Mohamed, Julien, Vincent, Runwey and Christina, for the pleasant coffee breaks and the many TA-ing and skiing adventures. I was very lucky to share the office with Lyudmila who is by now a dear friend with whom I can discuss any personal or work-related matters and on whose tea-and-sweets company I could always count. Thanks to Farnood, for his enthusiasm for research, food, sport (and more) and his most unpredictable questions ranging from German vocabulary to the evolution of religious dogmas. I owe much to Lucas who is a dear friend and a brilliant researcher: thank you for patiently answering all my questions, for showing me the beauty and the

benefits of thoroughly analyzing seemingly fastidious problems and for providing very useful feedback on my research results. I thank Victor for his curiosity about and beyond computer science matters and for making me a bit more environmentally aware; William for the taste for good life and for being an enthusiastic and hard-working collaborator; and Sébastien for his relaxed, positive and encouraging perspective.

I was lucky to be supported by many friends, nearby and far away, who are really too many to be able to mention them all...

I deeply enjoyed the company of Andrei, Nastya and Adrian with whom I shared many mountain adventures and delicious dinners and who gave me a great encouragement in the final steps of writing this thesis, thank you for your hospitality and generosity. Thank you, Alevtina, for your fun company and for being such a strong person and a dear friend. Thank you Serj and Sahar for your support and hospitality.

Among my Italian friends, special thanks go to Federica for her calm and constant advice and for insisting that I should trust my heart. Thanks to all the Zoccoli Duri for always keeping an open door for me whenever I returned to Milan and for always showing interest in all my difficulties and progresses. Thank you, Caterina, Paolo, Francesco, Nicola & Anna, Laura & Giacomo, Lidia & Geoffreoy, for all your patient support and encouragement.

Thanks to all the friends I met in Lausanne who really were like a second family to me: the Foletti, Franscini, Averchi, Ruggiero, Coppa, Anabela, Sara, Francesco and many more. Thank you, my many colocataires, Manu, Giulia R., Elena, Chiara, Marta, Giulia C., Ilaria and Pauline, for always being there to support me and listen to my stories.

Thank you, Young-Jun, for all the fun we have together and for your company through all the joys and sorrows of these years; thank you for your modesty and generosity, and for all the encouragement you give me.

Finally, thanks to my beloved family, mamma, papà, Guido and Sara, who are always there for me in spite of their own difficulties and to whom this thesis is dedicated. You always silently remind me that, in the end, what really matters is not what we accomplish but who we are.

Lausanne, February 2018

B. M. S.

Abstract

Localizing the source of an epidemic is a crucial task in many contexts, including the detection of malicious users in social networks and the identification of "patient zeros" of disease outbreaks. The difficulty of this task lies in the strict limitations on the data available: In most cases, when an epidemic spreads, only few individuals, who we will call *sensors*, provide information about their state. Furthermore, as the spread of an epidemic usually depends on a large number of variables, accounting for all the possible spreading patterns that could explain the available data can easily result in prohibitive computational costs. Therefore, in the field of source localization, there are two central research directions: The design of practical and reliable algorithms for localizing the source despite the limited data, and the optimization of data collection, i.e., the identification of the most informative sensors. In this dissertation we contribute to both these directions.
We consider network epidemics starting from an unknown source. The only information available is provided by a set of sensor nodes that reveal if and when they become infected. We study how many sensors are needed to guarantee the identification of the source. A set of sensors that guarantees the identification of the source is called a double resolving set (DRS); the minimum size of a DRS is called the double metric dimension (DMD). Computing the DMD is, in general, hard, hence estimating it with bounds is desirable. We focus on $\mathcal{G}(N,p)$ random networks for which we derive tight bounds for the DMD. We show that the DMD is a non-monotonic function of the parameter p, hence there are critical parameter ranges in which source localization is particularly difficult.

Again building on the relationship between source localization and DRSs, we move to optimizing the choice of a fixed number K of sensors. First, we look at the case of trees where the uniqueness of paths makes the problem simpler. For this case, we design polynomial time algorithms for selecting K sensors that optimize certain metrics of interest. Next, turning to general networks, we show that the optimal sensor set depends on the distribution of the time it takes for an infected node u to infect a non-infected neighbor v, which we call the *transmission delay* from u to v. We consider both a low- and a high-variance regime for the transmission delays. We design algorithms for sensor placement in both cases, and we show that they yield an improvement of up to 50% over state-of-the-art methods.

Finally, we propose a framework for source localization where some sensors (called *dynamic* sensors) can be added while the epidemic spreads and the localization progresses. We design an algorithm for joint source localization and dynamic sensor placement; This algorithm can handle two regimes: *offline* localization, where we localize the source after the epidemic spread, and *online* localization, where we localize the source while the epidemic is ongoing. We conduct an empirical study of offline and online localization and show that, by using dynamic sensors, the number of sensors we need to localize the source is up to 10 times less with respect to a strategy where all sensors are deployed *a priori*. We also study the resistance of our methods to high-variance transmission delays and show that, even in this setting, using dynamic sensors, the source can be localized with less than 5% of the nodes being sensors.

Key words: Source localization; Network epidemics; Sensor placement.

Riassunto

Localizzare il punto sorgivo di un'epidemia è un problema cruciale in diversi contesti, per esempio l'individuazione di soggetti che utilizzano in modo improprio i social networks oppure l'identificazione dei pazienti-zero nella diffusione di una malattia infettiva. Le strette limitazioni nella disponibilità di dati rendono questo problema difficile: Nella maggior parte dei casi, quando un'epidemia si propaga, solo alcuni individui, che chiamiamo qui *sensori*, forniscono informazioni riguardo al loro stato. Inoltre, poiché la diffusione di un'epidemia dipende spesso da molte variabili, tener conto di tutti i possibili scenari che potrebbero rendere ragione delle informazioni disponibiili porta spesso a costi computazionali proibitivi. Quindi, nella ricerca sulla localizzazione del punto sorgivo di un'epidemia, due direzioni sono centrali: il design di algoritmi applicabili in pratica e affidabili che possano localizzare la sorgente nonostante i dati a disposizione siano limitati e l'ottimizzazione della raccolta dei dati, ossia l'identificazione dei sensori più informativi. In questa tesi diamo un contributo in entrambe queste direzioni.

Dapprima studiamo quanti sensori sono necessari per garantire l'identificazione della sorgente. Un insieme di sensori con cui è possibile garantire l'identificazione della sorgente è un *double resolving set* (DRS); il minimo numero di nodi in un DRS si chiama *double metric dimension* (DMD). Calcolare la DMD è, in generale, difficile. Quindi è desiderabile poterne fare delle stime. Ci interessiamo a reti casuali di tipo $\mathcal{G}(N, p)$ per le quali deduciamo una stima asintoticamente precisa per la DMD. In questo modo mostriamo che la DMD è una funzione non monotòna del parametro p e che ci sono quindi dei valori critici di p per i quali localizzare la sorgente è sostanzialmente più difficile.

Utilizzando ancora il legame tra la localizzazione della sorgente e i DRS, passiamo poi a ottimizzare la scelta di un numero prefissato K di sensori. Per prima cosa, studiamo il caso particolare degli alberi in cui l'unicità del cammino tra qualsiasi coppia di nodi rende il problema più semplice. In questo caso, proponiamo algoritmi che, in tempo polinomiale, selezionano K sensori che ottimizzano alcune misure rilevanti. Successivamente, passando a reti generali, mostriamo che l'insieme ottimale di sensori dipende dalla distribuzione dei *tempi di trasmissione* da nodo infetto a nodo sano. Consideriamo sia il caso di alta che quello di bassa variabilità dei tempi di trasmissione, proponiamo algoritmi per la scelta di sensori in entrambi i casi e otteniamo un miglioramento fino al 50% rispetto ai metodi

correntemente in uso.
Infine, definiamo un modello più generale per la localizzazione della sorgente, in cui alcuni sensori, detti dinamici, possono essere aggiunti mentre l'epidemia si diffonde e la localizzazione della sorgente si precisa. Proponiamo un algoritmo con cui è possibile, allo stesso tempo, localizzare la sorgente e scegliere dei sensori dinamici. Questo algoritmo funziona sia quando si vuole localizzare la sorgente dopo la diffusione di un'epidemia (in modo offline) sia quando si vuole localizzare la sorgente di un'epidemia che è in corso (in modo online). Con un accurato studio sperimentale (del modo offline e del modo online) mostriamo che, usando sensori dinamici, il numero di sensori necessari per localizzare la sorgente è fino a 10 volte meno rispetto a metodi in cui tutti i sensori sono scelti a priori. Inoltre, per quanto riguarda la resistenza del nostro approccio all'alta variabilità dei tempi di trasmissione, mostriamo che, anche in questa condizione, usando sensori dinamici è possibile localizzare la sorgente scegliendo meno del 5% dei nodi come sensori.

Parole chiave: Localizzazione della sorgente; Epidemie su reti; Scelta di sensori.

Contents

Acknowledgements **i**

Abstract **iii**

1 Introduction **1**
- 1.1 Outline and Contributions . . . 3
- 1.2 Model and First Observations . . . 5
  - 1.2.1 Model . . . 5
  - 1.2.2 Localization Based on Relative Distances . . . 8
  - 1.2.3 Metrics for Source Localization . . . 11
  - 1.2.4 Sensor Placement . . . 13
- 1.3 Main Related Work . . . 16
  - 1.3.1 Epidemic Models . . . 16
  - 1.3.2 Observation Settings . . . 17
  - 1.3.3 Sensor Placement . . . 21
  - 1.3.4 Online Sensor Placement . . . 22
  - 1.3.5 Other Related Work . . . 23

2 Double Metric Dimension (DMD) of Random Networks **25**
- 2.1 Overview . . . 26
- 2.2 Preliminaries . . . 27
  - 2.2.1 The Metric Dimension (MD) . . . 27
  - 2.2.2 Contrast between the MD & the DMD . . . 31
  - 2.2.3 Definitions & Useful Lemmas . . . 32
- 2.3 Upper Bounds on the DMD . . . 34
  - 2.3.1 Bound for $\mathcal{G}(N,p)$ with $p = \Theta(1)$ . . . 35
  - 2.3.2 Bound for $\mathcal{G}(N,p)$ with $p = o(1)$ . . . 36
  - 2.3.3 Overall Bound on the DMD for $\mathcal{G}(N,p)$ . . . 44
  - 2.3.4 Expansion Properties of Random Networks . . . 45
- 2.4 Experimental Results . . . 49

2.4.1 The DMD of $\mathcal{G}(N, p)$ Networks 49
2.4.2 The DMD and the MD of Other Random Networks 51
2.5 Discussion 53

3 Sensor Placement on Trees 57
3.1 Overview 57
3.2 Preliminaries 58
3.2.1 Model 58
3.2.2 Noise Tolerance 59
3.2.3 Resolved Nodes 60
3.3 Success Probability Maximization 61
3.4 Expected Error Distance Minimization 63
3.5 Extensions 66
3.5.1 Weighted Nodes 67
3.5.2 Non-uniform Priors 67
3.6 Discussion 68

4 Sensor Placement on General Networks: The Effect of the Transmission-Delays Variance 69
4.1 Overview 69
4.2 Preliminaries 72
4.2.1 Model 72
4.2.2 Source Localization 72
4.2.3 Metrics 73
4.3 The Low-variance Regime 73
4.3.1 Noise Tolerance 74
4.3.2 Algorithm for Sensor Placement 75
4.3.3 The Approximating Algorithm of [Chen et al., 2014] 76
4.3.4 Comparison with Benchmarks 77
4.4 The High-variance Regime 78
4.4.1 Estimation of the Source 79
4.4.2 Algorithm for Sensor Placement 81
4.5 Experimental Results 85
4.5.1 Datasets 85
4.5.2 Comparison with Benchmarks 88
4.5.3 Distributions for the Transmission Delays 88
4.5.4 Evaluation of the Probability of Success and of the Expected Error Distance 89
4.6 Discussion 91

5 Sensor Placement on General Networks: Static Vs Dynamic 93
5.1 Overview . . . 93
5.2 Preliminaries . . . 96
  5.2.1 Model . . . 96
  5.2.2 Online & Offline Source Localization . . . 97
  5.2.3 Sensors . . . 97
5.3 Offline Localization with Static Sensors (S-OFF) . . . 99
  5.3.1 Deterministic Epidemics . . . 101
  5.3.2 Non-deterministic Epidemics . . . 102
5.4 Online Localization with Static Sensors (S-ON) . . . 104
5.5 Offline Localization with Static and Dynamic Sensors (D-OFF) . . . 106
  5.5.1 Correctness . . . 107
  5.5.2 Natural Gain Functions . . . 108
5.6 Online Localization with Static and Dynamic Sensors (D-ON) . . . 110
  5.6.1 Correctness . . . 110
5.7 Experimental Results . . . 111
  5.7.1 Experimental Setup . . . 111
  5.7.2 Network Topologies . . . 112
  5.7.3 Choice of the Static Sensors . . . 115
  5.7.4 Online vs Offline Localization . . . 115
  5.7.5 Static vs Dynamic Localization . . . 117
  5.7.6 Dynamic Sensors: How to Choose, When to Deploy and How Many 119
  5.7.7 Number of Candidate Sources at Successive Steps . . . 123
  5.7.8 Comparison with Existing Methods . . . 124
  5.7.9 Resistance to Unbounded Delay Distributions . . . 125
5.8 Additional Technical Details . . . 125
  5.8.1 S-OFF . . . 125
  5.8.2 S-ON . . . 129
  5.8.3 D-ON . . . 132
  5.8.4 Extending the Gain Functions to Negative Observations . . . 133
  5.8.5 Approximate SIZE-GAIN for the Non-deterministic Case . . . 135
5.9 Discussion . . . 137

Conclusion 139

Bibliography 149

Curriculum Vitae 151

1 Introduction

The phenomena that can be represented as a set of entities (or individuals) interacting and communicating between them are countless. Examples span from a set of individuals, who are connected if they can exchange information, to computers that are interconnected in a computer network; and from train networks, where two cities are connected if they can be reached in a single-hop journey, to hydraulic systems, where two points are connected if water can flaw directly between them.

In a mathematical language, each of these phenomena can be modeled as a *network*: a set of objects (which we call *nodes*) together with a set of connections between pairs of them (which we call *edges*). Formally, if $\mathcal{G}$ is a network, we write $\mathcal{G} = \mathcal{G}(V, E)$ where V is the set of nodes in $\mathcal{G}$ and E is its set of edges. An edge is a pair uv such that $u, v \in V$, and $uv \in E$ means that u is connected to v in $\mathcal{G}$.

In many applications, having full knowledge of the network of interest is not easy. The difficulties start with the definition of the relevant network for the problem to study and also arise from privacy concerns and, importantly, from to the cost of gathering data, which in turn results in open-access issues because having access to the network can be a competitive advantage. The problem of defining, reconstructing or approximating a network is actively studied from many perspectives: these span from using digital mobility traces to infer human contacts [Salathé et al., 2012], to reconstructing an online social network based on small subnetworks [Bollobás, 1990, Mossel and Ross, 2015, Yartseva et al., 2016] or based on observed sequences of events within the node population [Gomez Rodriguez et al., 2010, 2011] .

If a network, or an approximation of it, is given, it can be used to study diffusion phenomena, which we call *epidemics*. During an epidemic, an information or infection is passed through the network from node to node. When we study the spread of an epidemic $\mathcal{I}$, we say that a node becomes *infected* when it is reached by $\mathcal{I}$. Here, $\mathcal{I}$ can represent an infectious disease spreading in a population, e.g., a rumor being passed in a social network, an environment contamination or a computer worm. We call the time required by a node u to pass $\mathcal{I}$ to a neighbor z the *transmission delay* between u and z.

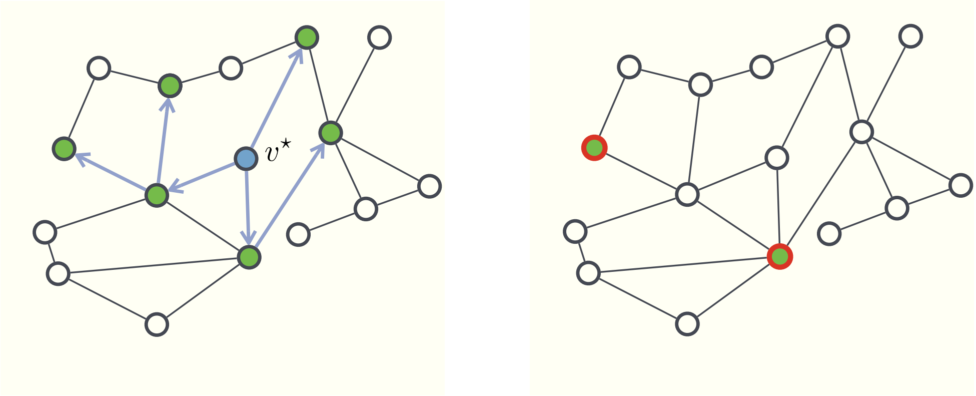
Many problems regarding network epidemics have been extensively studied. Some examples concern the extinction time of an epidemic and the conditions under which an epidemic persists in a network [Berger et al., 2005, Nowzari et al., 2016]; the strategies through which an epidemic can be mitigated or extinguished, for example, through vaccine allocation [Preciado et al., 2013, Drakopoulos et al., 2014, Scaman et al., 2016] or network rearrangement [Tong et al., 2012]; the identification of the most influential nodes, i.e., the ones that maximize the epidemic spread [Richardson and Domingos, 2002, Kempe et al., 2003, Leskovec et al., 2007a] (the latter problem is commonly called *viral marketing*).

Another difficult and intriguing task, which we address in this dissertation, deals with localizing the *source of an epidemic*, i.e., the first node that became infected, hence it is the origin of the epidemic. Detecting the origin of a worm in a computer network, identifying a false-rumor instigator in a social network or finding the patient-zero of a virulent disease can be crucial for both the containment of an ongoing epidemic and for the prevention of future outbreaks. Focusing on a case in which the epidemic is originated by a single source, we denote the source by $v^\star \in V$. In source localization, we want to produce an estimator $\widehat{v}$ for $v^\star$ such that $\widehat{v}$ is *sufficiently close* to $v^\star$ or, in a slightly different formulation, we want to find a small set $\mathcal{B}$ of nodes such that, with high-probability, $v^\star \in \mathcal{B}$.

It is clear that if it we could completely observe an epidemic propagation, identifying its source would be very easy. Unfortunately, due to the costs of information collection and to overhead constraints, the data available for source localization is often very sparse: The number of nodes in the network is often prohibitively large and some of them can be unable or unwilling to provide information about their state. When studying an infectious disease, for example, performing the necessary medical exams and data analysis on many suspected households or communities can be very expensive, whereas the efficient allocation of resources can lead to enormous savings [Somda et al., 2009].

We refer to the network nodes whose state is observed as *sensors*. We are interested, in particular, in how these sensors must be chosen in order to optimize source localization. In other words, calling $\mathcal{O} = \mathcal{O}(\mathcal{U})$ the set of observations obtained using the sensor set $\mathcal{U} \subseteq V$ and considering a metric $\mathcal{M}$ for the performance of source localization, we are interested in the choice of $\mathcal{U}$ that optimizes $\mathcal{M}(\mathcal{O}, v^\star)$. An example of a metric $\mathcal{M}$ is the probability of success in source localization, i.e., $\mathcal{P}_s(\mathcal{O}, v^\star) \overset{\text{def}}{=} \mathbf{P}(\widehat{v} = v^\star | \mathcal{O})$.

The difficulties faced in finding optimal sensor sets for source localization are two-fold. First, computing the likelihood of a node being the source, conditioned on the available observations, can be computationally prohibitive [Shah and Zaman, 2011, Pinto et al., 2012]; evaluating the probability of correct localization given a set of sensors is, in general, even harder. Second, the optimal selection of sensors is NP-hard, even when the



(a) Epidemic propagation (b) Available observations

Figure 1.1 – **(a)**: During an epidemic, an information or infection is passed through the network from node to node starting from an unknown source $v^\star$; **(b)**: The information that can be used to localize the source is usually very limited, e.g., only the states and infection times of a few sensors (marked with a red line) are known, and nothing is known about the other nodes.

transmission delays are deterministic (see Section 1.2.4).

In this dissertation we present innovative and efficient methods for sensor placement for source localization; our methods benefit from two essential features. First, in contrast to many common approaches, these methods are specifically designed for the source localization problem, i.e., the choice of the sensors is directly related to the way in which the collected information is used to localize the source. Second, they are applicable in a wide range of settings: For example, we study how to fit different levels of variance in the transmission delays or how to choose the sensors when they can be adaptively selected while the epidemic spreads and the localization process progresses.

1.1 Outline and Contributions

In Section 1.2 we describe the general setting we assume and we explain the relationship of source localization and of sensor placement with the double-resolving-set (DRS) problem. In this context, we also study the case of trees by presenting some first results and examples that will be useful in the following.

In Section 1.3, we survey the relevant related work. In the last few years an impressive amount of contributions to source localization have been published. We describe the main lines of approach, pointing out the different assumptions and the results achieved. We also position source localization in a wider spectrum of research that investigates similar research questions.

The remainder of the dissertation is organized in four chapters that present our four main contributions.

In Chapter 2 we look at the minimum number of sensors with which, on a given network, the identification of the source can, under certain hypothesis, be guaranteed. This minimum number is the minimum size of a DRS of the network, which we call the double metric dimension (DMD) of the network. We derive asymptotic bounds for the DMD of $\mathcal{G}(N, p)$ random networks when the parameter $p = p(N)$ varies in a wide range. Our bound undergoes a non-monotonic behavior as a function of p. We experimentally show that, indeed, the DMD of $\mathcal{G}(N, p)$ random networks is non-monotonic in p, meaning that we can identify parameter regimes where localizing the source is substantially more difficult than in others.

Chapter 3 and Chapter 4 are concerned with optimizing the choice of sensors for when all available sensors must be chosen *a priori*, i.e., independently of any epidemic instance.

In Chapter 3, we study the special case of trees where the uniqueness of paths makes the problem of sensor placement simpler. For this case, we give polynomial-time dynamic-programming algorithms to select the K sensors that optimize the probability of success in source localization and the expected distance between the real source and the estimated source. We also show how our algorithms are amenable for extensions to interesting variants of the problem, such as for when sensors can have different costs or for when a non-uniform prior on the identity of the source is available.

In Chapter 4, turning to general networks, we show that the optimal sensor set depends not only on the topology of the network, but also on the variance of the node-to-node transmission delays. We consider both a low-variance regime and a high-variance regime for the transmission delays and, in both cases, we propose algorithms for sensor placement. We prove that in the low-variance regime, it suffices to consider only the network-topology and to minimize the amount of nodes that appear as equivalent from the point of view of the sensors. However, the high-variance regime requires a different approach in order to guarantee that the observed infection times are sufficiently informative about the location of the source and do not get masked by the noise in the transmission delays. This is accomplished by additionally ensuring that the sensors are not placed too far apart from each other. By simulating epidemics in three real-world networks, we show that, compared to state-of-the-art strategies for sensor placement, our methods have a better performance in terms of source localization accuracy for both the low- and the high-variance regimes.

In Chapter 5, we propose a general framework for source localization that incorporates two possible scenarios: In the first, as in Chapter 3 and Chapter 4, all sensors are chosen *a priori* independently of any epidemic process; in this case, we say that all sensors are *static*. In the second scenario, more sensors (called *dynamic* sensors) can be added as the epidemic spreads and the localization process progresses. We propose an algorithm for source localization and dynamic sensor placement; this algorithm can handle both the regime where we want to localize the source after the epidemic spread through the entire network (*offline* localization) and the one where we want to localize the source while the epidemic is still ongoing (*online* localization), in which case the information at our disposal is not only sparse but also incomplete. By experimenting with synthetic and real-world networks, we compare the performance of offline and online source localization by using both static sensors and dynamic sensors. We highlight how the trade-off between cost (in number of sensors) and precision (in source localization) emerges across these four variants. Moreover we study the impact of several model parameters such as the relative number of static and dynamic sensors and the time delay between two successive deployments of dynamic sensors. Our analysis shows that, by using dynamic sensors, we dramatically outperform a static strategy with the same budget and that, even with high-variance transmission delays, the source can be localized with extremely few sensors, especially in real-world networks. Interestingly, we observe a switch in the optimal placement of the static sensors when the variance of the transmission delays increases, which is consistent with the results of Chapter 4.

1.2 Model and First Observations

In this section, we introduce the main notation, definitions and preliminary results that are used throughout this manuscript. In Section 1.2.1, we describe the general network-diffusion model we adopt. In Section 1.2.2, we introduce the notion of a double resolving set in relation to source localization. In Section 1.2.3, we define and compare several metrics of interest for source localization. Finally, in Section 1.2.4 we present some first results for tree networks and prove a hardness result for sensor placement on general networks, which motivates our search for scalable approaches to sensor placement on general networks.

1.2.1 Model

Network

We model a set of contacts with a weighted network $\mathcal{G}(V,E)$ with $|V| = N$. For every edge $uv \in E$, the weight $w_{uv} \in \mathbb{R}^+$ is the average time it takes for an infection to spread from u to v. $\mathcal{G}$ is undirected, i.e., $w_{uv} = w_{vu}$ for every $uv \in E$. If $u, v \in V$ are two distinct nodes, a path connecting u and v is a sequence of k edges ($k \in \mathbb{N}^+$) of the form $u_{1,1}u_{1,2}, .., u_{k,1}u_{k,2}$ where $u_{1,1} = u$, $u_{k,2} = v$ and, for every $i \in \{1, .., k-1\}$, $u_{i,2} = u_{i+1,1}$. The distance $d(u, v)$ between two nodes u and v is the minimal sum of edge weights along a path connecting u and v. When the path connecting two nodes u, v is unique, we denote it with $\mathcal{P}(u, v)$; i.e., $\mathcal{P}(u, v)$ is a sequence of nodes $\{v_1, .., v_{m+1}\}$ where $v_1 = u$,

Notation

$\mathbb{N}$ (resp., $\mathbb{N}^+$)	positive integers including (resp., excluding) 0
$\mathcal{G}(E, V)$	contact network
$N = \lvert V \rvert$	network size
w_{uv}	weight of edge uv ($\in \mathbb{R}^+$)
X_{uv}	random transmission delay on edge uv
$d(x, y)$	weighted distance between x and y ($\in \mathbb{R}^+$)
$\mathcal{P}(x, y)$	path (sequence of nodes) from x to y (when unique)
$v^\star$	unknown source
$t^\star$	unknown starting time of the epidemic
t_u	infection time of node u
$T(v, u)$	$t_u - t^\star$, random infection delay of node u when $v^\star = v$
$\mathcal{U}$	set of sensors
K	budget for sensors
$\mathcal{O}$	set of observations

$v_2 = v$ and m is the number of edges in $\mathcal{P}(u, v)$.

Epidemic

An epidemic starts from an unknown *single source* at an *unknown time* $t^\star$. An extension of our results to the case of multiple sources could use the work by Zhang et al. [2015] on a related problem, but it is not addressed in this thesis.[1]

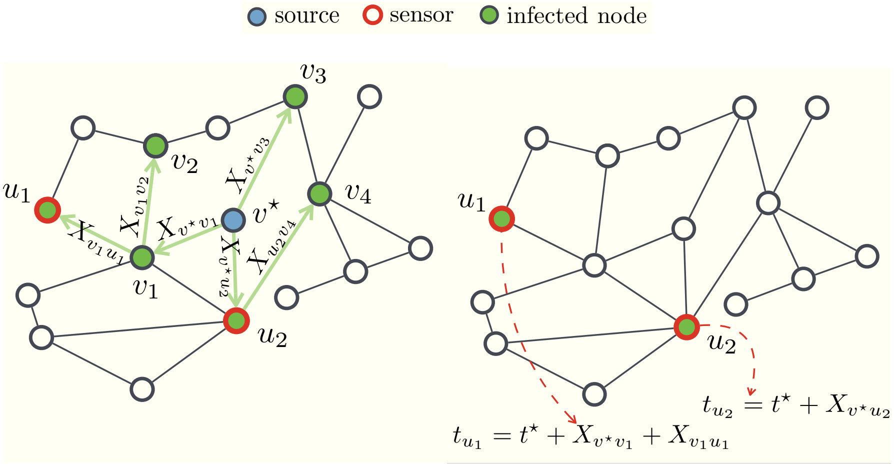
The identity of the source is an unobserved random variable $v^\star$ that takes values in the node set V. We denote the prior distribution on the identity of the source with π. Unless otherwise specified, we assume that we have no prior knowledge on the identity of the source, hence we take π to be the uniform distribution over the nodes in V.

We use the SI epidemic model adopted, among others, by Pinto et al. [2012] and Luo and Tay [2012]: At any time, every node is in one of two possible states, S (*Susceptible*) or I (*Infected*). Nonetheless, as our methods for source localization only use the time at which the sensors are first infected (no assumption on recovery or re-infection dynamics is made), they can be applied to any epidemic model, including SIS or SIR (provided that nodes do not recover before infecting their neighbors).

For every edge $uv \in E$, let X_{uv} be the time it takes for an infection to spread from u to v. X_{uv} is called the *transmission delay* on edge uv and is a positive random variable with

---

[1] Zhang et al. [2015] extended the definition of double resolving sets to set-resolving sets (SRS): the idea is that, based on the distances to the nodes in a SRS, it is possible to distinguish amongst different possible groups of coexisting sources.



(a) Epidemic propagation (b) Available observations

Figure 1.2 – **(a)**: An epidemic propagates in a network starting from a single unknown source; **(b)**: Only the sensors provide information about the epidemic process by revealing their infection time.

mean $\mathbf{E}[X_{uv}] = w_{uv}$. Denote by t_u the infection time of a node u, i.e., the first time u is reached by the epidemic. An infected node u infects any non-infected neighbor v at time $t_v = t_u + X_{uv}$. We assume that the variables $\{X_{uv}\}_{uv \in E}$ are mutually independent. This model implies that all nodes eventually become infected. When $v^\star = v$, we denote by $T(v, u)$ the total time it takes for the infection to spread from v to a node $u \in V$.

In Chapters 3-5, we will consider different distributions for the variables $\{X_{uv}\}_{uv \in E}$ that will be specified in the relevant chapters. In particular we will often distinguish the *deterministic* case where $X_{uv} = \mathbf{E}[X_{uv}] = w_{uv}$ for every $uv \in E$ ($\mathrm{Var}(X_{uv}) = 0$) from a *noisy* (or *non-deterministic* case) where $\mathrm{Var}(X_{uv}) > 0$.

Sensors

We localize the source by using the information provided by a subset of nodes that we call sensors.

Definition 1.1 (Sensor)**.** *A sensor is a node that can reveal its infection state (S or I) and, if it is infected, its infection time.*

Note that if $v^\star = v$ and v is a sensor, v provides information in the same way as any other sensor i.e., it only reveals, if infected, its infection time (which would be equal to $t^\star$), but it does not reveal itself to be the source.

1.2.2 Localization Based on Relative Distances

If an epidemic spreads deterministically starting from $v^\star = v$, then $T(v,u) = t_u - t^\star = d(v,u)$ for every $u \in V$. If the spread is non-deterministic, the same relation holds in average, i.e., $\mathbf{E}[T(v,u)] = d(v,u)$, when the path connecting v and u is unique.[2] Hence if $t^\star$ is known, $T(v,u)$ can be interpreted as a proxy for $d(v^\star,u)$ and the infection time t_u can be directly used to localize the source.

However, as we assume that $t^\star$ is unknown, we cannot use the infection time of a single sensor to infer the identity of the source. Instead, we use the *differences* between the infection times of pairs of sensors. If the sensor set is $\mathcal{U}$, we use the differences $\{t_u - t_z\}_{u,z\in\mathcal{U}}$. Borrowing the terminology used for the localization of transmitting devices, our work is a TDOA (time difference of arrivals) approach to source localization (in contrast with a TOA approach where the time of arrivals - and the starting time $t^\star$ - are used)[Li et al., 2016].

Consider now the case of a deterministic epidemic and two possible sources v_1 and v_2. We can distinguish which of the nodes is the source using the set $\{t_u - t_v\}_{u,v\in\mathcal{U}}$ if and only if there exist $u_1, u_2 \in \mathcal{U}$ such that

$$d(u_1,v_1) - d(u_1,v_2) \neq d(u_2,v_1) - d(u_2,v_2).$$

Definition 1.2 (Distinguished nodes). *Let $\mathcal{U} \subseteq V$ with $|\mathcal{U}| \geq 2$. A node v_1 is distinguished from a node v_2 by $\mathcal{U}$ if and only if there exist $u_1, u_2 \in \mathcal{U}$ such that*

$$d(u_1,v_1) - d(u_1,v_2) \neq d(u_2,v_1) - d(u_2,v_2). \tag{1.1}$$

In this case, we say that v_1, v_2 are distinguished by the pair u_1, u_2.

Definition 1.3 (Equivalent nodes). *Let $\mathcal{U} \subseteq V$ with $|\mathcal{U}| \geq 2$. A node v_1 is said to be equivalent to a node v_2 with respect to $\mathcal{U}$, (which we write $v_1 \sim v_2$) if and only if, for every $u_1, u_2 \in \mathcal{U}$*

$$d(u_1,v_1) - d(u_1,v_2) = d(u_2,v_1) - d(u_2,v_2). \tag{1.2}$$

The relation $\sim$ of Definition 1.3 is reflexive, symmetric, and transitive, hence it defines an *equivalence relation.* Therefore, a set of sensors $\mathcal{U}$ partitions V in *equivalence classes* (an example is given in Figure 1.3). We denote by $[v]_{\mathcal{U}}$ the class of v, i.e., the set of all nodes that are equivalent to v. When the identity of the sensor set $\mathcal{U}$ is clear from the context, we simply write $[v]$.

---

[2] If the path connecting v and u is not unique, we have the bound $\mathbf{E}[T(v,u)] \leq d(v,u)$ and this bound becomes looser when there are many alternative paths connecting u and v whose length is similar to $d(u,v)$.

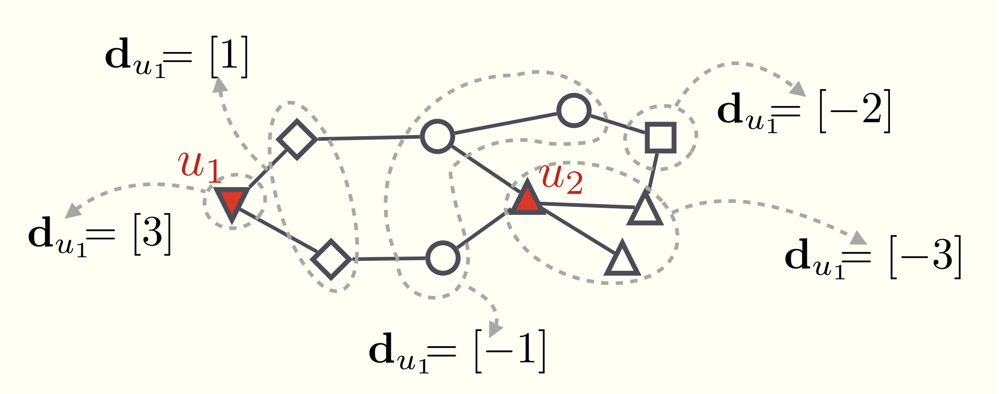


Figure 1.3 – A network with two sensors u_1 and u_2 and all edge weights equal to 1. Groups of nodes which are equivalent with respect to the red sensors are grouped by dotted lines and represented by different node shapes. In this example there are 5 equivalence classes.

Double Resolving Sets

A set $\mathcal{Z}$ such that for every $v_1, v_2 \in V$, v_1 and v_2 are distinguished by $\mathcal{Z}$ is called a *double resolving set* (DRS) of $\mathcal{G}$. More formally, we have the following definitions.

Definition 1.4 (Double resolving set)**.** *Let $\mathcal{U} \subseteq V$, $|\mathcal{U}| \geq 2$. $\mathcal{U}$ is a double resolving set for $\mathcal{G}$ if and only if, for every $v_1, v_2 \in V$ there exist u_1, u_2 in $\mathcal{U}$ such that*

$$d(u_1, v_1) - d(u_1, v_2) \neq d(u_2, v_1) - d(u_2, v_2).$$

Definition 1.5 (Double metric dimension)**.** *The double metric dimension (DMD) of a network $\mathcal{G}$ is the minimum size of a double resolving set of $\mathcal{G}$.*

The DMD of a network is, by definition, larger or equal than 2. In relation to source localization, the DMD of a network is equal to the *minimum number of sensors* needed to identify the source of a deterministic epidemic among any set of possible sources.

The problem of finding the minimum-size DRS (hence the DMD) of a network is known as the *minimum-double-resolving-set problem* (MDRS) [Cáceres et al., 2007]. Although solving MDRS is, in general, NP-hard [Kratica et al., 2009], it is easy to compute the DMD and to find a minimum size DRS for some particular network topologies such as cycles, wheels and k-augmented trees [Chen et al., 2014]. The next proposition, proven differently by Chen et al. [2014], gives a result for trees that we will use in the next chapters.

Proposition 1.6 (DMD of trees. Lemma 4.1 in [Chen et al., 2014])**.** *Let $\mathcal{T} = \mathcal{T}(V, E)$ be a tree and let $\mathcal{L}$ the set of all the leaves of $\mathcal{T}$. $\mathcal{U} = \mathcal{L}$ is the only minimum size* DRS *of $\mathcal{T}$ and* $\text{DMD}(\mathcal{T}) = |\mathcal{L}|$.

Proof. First, we prove that, if $\mathcal{U} = \mathcal{L}$, all equivalence classes are singletons. Let $v, z \in V$, $v \neq z$ and take $\ell_1, \ell_2 \in \mathcal{L}$ such that $v, z \in \mathcal{P}(\ell_1, \ell_2)$. Without loss of generality, we can assume that $d(v, \ell_1) < d(z, \ell_1)$ and $d(z, \ell_2) < d(v, \ell_2)$, i.e., going from ℓ_1 to ℓ_2, we first pass through v and then through z. Hence $d(z, \ell_1) - d(z, \ell_2) > d(v, \ell_1) - d(v, \ell_2)$ and we can conclude that $[v] \neq [z]$.

Second, we prove that if $\mathcal{L} \not\subseteq \mathcal{U}$, $\mathcal{U}$ is not a DRS of $\mathcal{T}$. Let $\ell \in \mathcal{L} \backslash \mathcal{U}$ and let v be the only neighbor of ℓ. For every $u \in \mathcal{U}$, $d(v, u) = d(\ell, u) - 1$. Hence, for every $u_1, u_2 \in \mathcal{U}$, $d(v, u_2) - d(v, u_1) = d(\ell, u_2) - d(\ell, u_1)$, i.e., $[v] = [\ell]$, and we conclude that $\mathcal{U}$ is not a DRS of $\mathcal{T}$. □

When proving that if $\mathcal{L} \not\subseteq \mathcal{U}$, $\mathcal{U}$ is not a DRS, we did not use that $\mathcal{T}$ is a tree. Hence we have the following corollary.

Corollary 1.7. *Let $\mathcal{G} = \mathcal{G}(V, E)$ be a network and let $\mathcal{L}$ be the set of all the leaves of $\mathcal{G}$. If $\mathcal{U}$ is a* DRS *of $\mathcal{G}$, then $\mathcal{L} \subseteq \mathcal{U}$.*

Although a few results concerning the minimum-size DRS of specific network topologies are available [Chen et al., 2014], finding a minimum-size DRS for general networks is NP-hard. A $(1 + o(1)) \log(N)$-approximation algorithm was proposed by Chen et al. [2014] (see also Section 4.3.3).

Note that the DMD of a network can be as large as $N - 1$ (e.g., for a complete network), and choosing the nodes forming a DRS as sensors can be, in many practical cases, prohibitively expensive (see Figure 1.4). Furthermore, this would guarantee that it is possible to localize the source only for deterministic epidemics. For this reason, studying how to allocate a limited number of sensors in order to guarantee a good performance of source localization is a crucial aspect of source localization.

Distance Vectors & Observation Vectors

We now define the *distance vector* of a candidate source.

Definition 1.8 (Distance vector). *Let $\mathcal{U} \subseteq V$ with $|\mathcal{U}| = K \geq 2$ be a set of sensors and let $u_1 \in \mathcal{U}$. For each candidate source $v \in V$ the distance vector of v (with respect to u_1) is $\mathbf{d}_{v,u_1} \in \mathbb{R}^{K-1}$ with entries $d(v, u_i) - d(v, u_1)$ for $2 \leq i \leq K$.*

The following lemma, shows that the equality between distance vectors of different candidate sources does not depend on the choice of the reference sensor u_1 of Definition 1.8.

Lemma 1.9 (Independence from the reference sensor. Lemma 3.1 in [Chen et al., 2014]). *Let $\mathcal{U} \subseteq V$ with $|\mathcal{U}| = K \geq 2$, $u_1 \in \mathcal{U}$ and let $v_1, v_2 \in V$. Then, $[v_1]_{\mathcal{U}} = [v_2]_{\mathcal{U}}$ if and only if $\mathbf{d}_{v_1,u_1} = \mathbf{d}_{v_2,u_1}$, for any choice of the reference sensor u_1.*

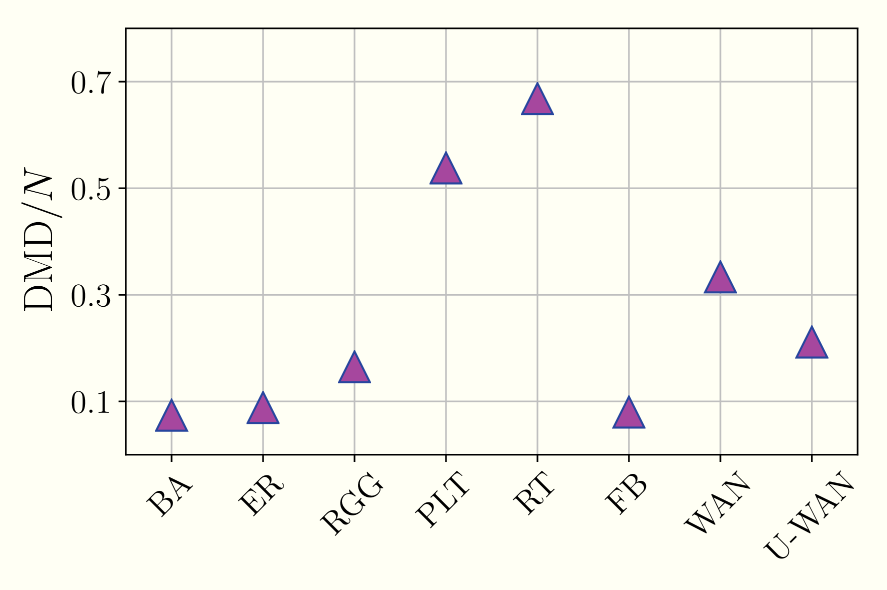


Figure 1.4 – Approximate DMD as a fraction of the network size for the networks presented in Section 5.7.2. The approximation is computed with the $(1+o(1))\log(N)$-approximation algorithm of Chen et al. [2014].

Proof. Let $u_1 \in \mathcal{U}$ be the reference sensor. ($\Rightarrow$) Assume $[v_1]_{\mathcal{U}} = [v_2]_{\mathcal{U}}$. Then, by definition, $d(v_1, u) - d(v_1, u_1) = d(v_2, u) - d(v_2, u_1)$ for all $u \in \mathcal{U}$, and hence $\mathbf{d}_{v_1,u_1} = \mathbf{d}_{v_2,u_1}$.

($\Leftarrow$) Assume conversely that $\mathbf{d}_{v_1,u_1} = \mathbf{d}_{v_2,u_1}$. Then, for every $u_i, u_j \in \mathcal{U}\backslash\{u_1\}$, both equalities $d(v_1, u_i) - d(v_2, u_i) = d(v_1, u_1) - d(v_2, u_1)$ and $d(v_1, u_j) - d(v_2, u_j) = d(v_1, u_1) - d(v_2, u_1)$ hold. Therefore, by transitivity, $d(v_1, u_j) - d(v_2, u_j) = d(v_1, u_i) - d(v_2, u_i)$ and the two nodes are equivalent, i.e., $[v_1]_{\mathcal{U}} = [v_2]_{\mathcal{U}}$. $\square$

Hence we base the localization of the source on a *vector of observations* where, again, the choice of a reference sensor can be made arbitrarily.

Definition 1.10 (Observation vector)**.** *Let $\mathcal{U} \subseteq V$, $|\mathcal{U}| = K \geq 2$ be a set of sensors, $u_1 \in \mathcal{U}$ a reference sensor and $\{t_u\}_{u\in\mathcal{U}}$ the infection times of the sensors for a given epidemic. Then we call the observation vector of the epidemic the vector $\mathbf{t}_{u_1} \triangleq [t_{u_2} - t_{u_1}, .., t_{u_K} - t_{u_1}] \subseteq \mathbb{R}^{K-1}$.*

Note that, if an epidemic spreads deterministically starting from $v^\star = v$, $\mathbf{t}_{u_1} = \mathbf{d}_{v,u_1}$.

1.2.3 Metrics for Source Localization

In this section, we define some possible metrics of interest for source localization, and we show that optimizing these metrics can require different sets of sensors.

For ease of exposition, we restrict ourselves to deterministic epidemics. In the deterministic regime, the partition into equivalence classes determines if the source can be localized: If $[v^\star]$ is a singleton, it is always possible to localize the source exactly based on the observed infection time; if it is not a singleton, we can only correctly identify the *class* to which $v^\star$ belongs and we produce an estimated source $\widehat{v} \in [v^\star]$ sampling from $\pi\Big|_{[v^\star]}$. In the following we will assume that no prior is available and we will take π to be uniform.

We will express the following metrics for the simple algorithm we just described. However, they can be used to evaluate the performance of any algorithm that outputs a single estimate $\widehat{v}$.

The *success probability* $\mathcal{P}_s$ is defined as $\mathbf{P}(\widehat{v} = v^\star)$. If q is the number of equivalence classes identified by a sensor set $\mathcal{U}$, for the algorithm described above we have

$$\begin{aligned}\mathcal{P}_s &= \sum_{[v]\subseteq V} \mathbf{P}(\widehat{v} = v^\star | v^\star \in [v])\mathbf{P}(v^\star \in [v]) \\ &= \sum_{[v]\subseteq V} \frac{1}{|[v]|} \cdot \frac{|[v]|}{n} = \frac{1}{n} \sum_{[v]\subseteq V} 1 = \frac{q}{n}. \qquad (1.3)\end{aligned}$$

Note that $\mathcal{P}_s = 1$ if and only if all equivalence classes are singletons, and that $\mathcal{P}_s$ depends exclusively on the number of equivalence classes (and not, for example, on their size).

A second metric of interest is the expected error distance $\mathcal{D}_e \overset{\text{def}}{=} \mathbf{E}[d(\widehat{v}, v^\star)]$. Again, for the algorithm above, $\mathcal{D}_e$ can be computed from the partition in equivalence classes:

$$\begin{aligned}\mathcal{D}_e &= \mathbf{E}[d(v^\star, \widehat{v})] \\ &= \sum_{s\in V} \mathbf{P}(v^\star = s) \sum_{v\in[s]} \mathbf{P}(\widehat{s} = v | v^\star = s) d(s, v) \\ &= \frac{1}{n} \sum_{s\in V} \frac{1}{|[s]|} \sum_{v\in[s]} d(s, v), \qquad (1.4)\end{aligned}$$

$\mathcal{D}_e = 0$ if and only if all equivalence classes are singletons. Moreover, Equation 1.4 shows that $\mathcal{D}_e$ directly depends on the average distance between the nodes in a same equivalence class. In our experimental evaluations, we also sometimes consider an expression for the hop-distance (instead of the weighted distance as in (1.4)).

Maximizing $\mathcal{P}_s$ (respectively, minimizing $\mathcal{D}_e$) we minimize the probability of $\widehat{v} \neq v^\star$ (resp., the average distance between $v^\star$ and $\widehat{v}$). Other natural metrics for source localization are the *worst-case* versions of these metrics, i.e., the *minimum probability of success* $\widehat{\mathcal{P}_s} \overset{\text{def}}{=} \min_{[s]\subseteq V} \mathcal{P}_{s|v^\star\in[s]}$ and the *maximum distance between* $\widehat{v}$ *and* $v^\star$, denoted by $\widehat{\mathcal{D}_e}$.

More formally, for the algorithm above

$$\widehat{\mathcal{P}_s} = \min_{[v]\subseteq V} \frac{1}{|[v]|}$$

and $\widehat{\mathcal{D}}_e$ as

$$\widehat{\mathcal{D}}_e = \max_{[s]\subseteq V} \max_{t,v\in[s]} d(t,v).$$

These last two metrics are relevant, for example, in adversarial settings (e.g., in the case of bio-warfare) where, if the sensors are known, the adversary would want to select the *worst* location for the source.

A last natural metric, which interpolates between average and worst-case metrics, is the *expected maximum distance* between the true and the estimated source, which we define as $\overline{\mathcal{D}}_e \overset{\text{def}}{=} \mathbf{E}_{v^\star}[\max(d(v^\star, \widehat{v}))]$. For the algorithm above we have

$$\overline{\mathcal{D}}_e = \mathbf{E}_{v^\star}[\max d(v^\star, \widehat{v})] = \sum_{s\in V} \frac{1}{n}\left(\max_{t\in[s]} d(s,t)\right).$$

1.2.4 Sensor Placement

We define the sensor placement problem as follows: Given a budget $K \in \mathbb{N}$, i.e., a number of sensors $K < N$, and a metric $\mathcal{M} = \mathcal{M}(\mathcal{U})$ that we want to optimize, find a set $\mathcal{U}_{\text{opt}} \subseteq V$, $|\mathcal{U}_{\text{opt}}| \leq K$ that optimizes $\mathcal{M}$.

Example: Sensor Placement on Trees

We now look at sensor placement in the particular case of trees. First, we prove that, if $\mathcal{G}$ is a tree, for any of the metrics of Section 1.2.3 the minimum-size optimal sets of sensors are contained in the leaf set. Then we demonstrate with an example that optimizing the five metrics presented in Section 1.2.3 can require different set of sensors [Celis et al., 2015].

Lemma 1.11. *Let $\mathcal{T} = \mathcal{T}(V,E)$ be a tree and $\mathcal{U} \subseteq V$ a set of sensors. Then the number of equivalence classes q can be computed as*

$$q = \left|\left\{v : v \in \mathcal{P}(u_1,u_2) \textit{ for any } u_1, u_2 \in \mathcal{U}\right\}\right|.$$

Proof. Let us define $S \triangleq \{v : v \in \mathcal{P}(u_1,u_2) \text{ for any } u_1, u_2 \in \mathcal{U}\}$.

First, we prove that for every $v, z \in S$ with $v \neq z$ we have $[v] \neq [z]$. Take $v, z \in S$ with $v \neq z$. There exist $u_1, u_2, u_3, u_4 \in \mathcal{U}$, with possibly u_1 (or u_2) equal to u_3 (or u_4) such that $v \in \mathcal{P}(u_1, u_2)$ and $z \in \mathcal{P}(u_3, u_4)$. Since $\mathcal{T}$ is a tree, there exist $i, j \in \{1, 2, 3, 4\}$ such that $v, z \in \mathcal{P}(u_i, u_j)$. Then, the statement can be proven similarly to the first part of the proof of Proposition 1.6.

Second, if $z \notin S$, we prove that there exists $v \in S$ such that $[z] = [v]$. If $z \notin S$, there exists a neighbor v of z such that $\mathcal{U}$ is contained in the subtree $\mathcal{T}_v$ rooted at v and not containing z. Call w the last common node to all paths $\{\mathcal{P}(z, u)\}_{u \in \mathcal{U}}$. We have that $w \in S$. With a technique similar to the second part of the proof of Proposition 1.6, we can prove that $[z] = [w]$. □

Proposition 1.12. *Let $\mathcal{T} = \mathcal{T}(V, E)$ be a tree and let $\mathcal{L}$ be the set of all leaves of $\mathcal{T}$. Let K be the budget for sensors. If $\mathcal{U}$ is a set of minimum-size such that $\mathcal{U} \in argmax_{|\mathcal{U}| \leq K} \mathcal{P}_s(\mathcal{U})$, then $\mathcal{U} \subseteq \mathcal{L}$.*

Proof. Let $\mathcal{U}$ be a minimum-size sensor set that maximizes $\mathcal{P}_s$ among all possible sets with cardinality smaller than K. We know from (1.3) that $\mathcal{P}_s$ is maximized if and only if the number of equivalence classes is maximized.

We assume by contradiction that there exists $u \in \mathcal{U} \backslash \mathcal{L}$. We have two cases:

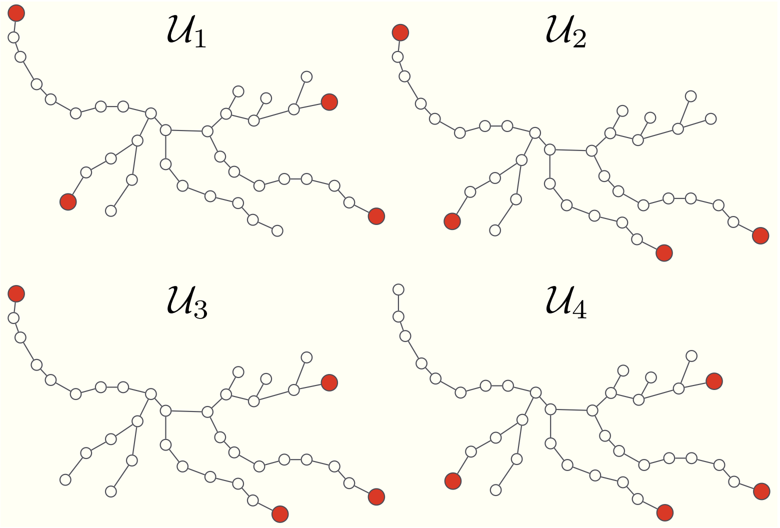
(a) There exist $u_1, u_2 \in \mathcal{U} \backslash \{u\}$ such that $u \in \mathcal{P}(u_1, u_2)$. In this case, we can apply Lemma 1.11 and say that if we remove u from $\mathcal{U}$ the value of $\mathcal{P}_s$ does not change. Hence, we have a contradiction with $\mathcal{U}$ being an optimal set of minimum size;

(b) There do not exist $u_1, u_2 \in \mathcal{U} \backslash \{u\}$ such that $u \in \mathcal{P}(u_1, u_2)$. In this case, there exists a subtree $\mathcal{T}_u$ rooted at u in which no leaf is a sensor. Let $\ell \in \mathcal{T}_u \cap \mathcal{L}$. If we substitute u with ℓ in $\mathcal{U}$, the number of nodes lying on the shortest path between two any sensor increases. Hence, applying again Lemma 1.11, we conclude that we found a set with the same cardinality of $\mathcal{U}$ but a highest value of $\mathcal{P}_s$, which gives a contradiction with the optimality of $\mathcal{U}$.

□

For the other metrics in Section 1.2.3 analogous results can be derived.

Consider now the tree in Figure 1.5, together with the four sets of $K = 4$ sensors represented in the four subfigures. Using the result of Proposition 1.12, only sensor sets contained in the leaf set can be optimal. We considered four of the possible sensor placements contained in the leaf set and having cardinality $K = 4$. These include, in particular, the placements that optimize $\mathcal{P}_s$, $\widehat{\mathcal{P}_s}$, $\mathcal{D}_e$, $\widehat{\mathcal{D}}_e$ and $\overline{\mathcal{D}}_e$. We see from the table in Figure 1.5 that the different metrics are optimized by different sensor sets.



	$\mathcal{P}_s$	$\widehat{\mathcal{P}_s}$	$\mathcal{D}_e$	$\widehat{\mathcal{D}}_e$	$\overline{\mathcal{D}}_e$
$\mathcal{U}_1$	0.71	**0.14**	0.57	6	0.26
$\mathcal{U}_2$	**0.76**	0.12	0.50	**4**	**0.16**
$\mathcal{U}_3$	**0.76**	**0.14**	**0.45**	5	0.18
$\mathcal{U}_4$	0.66	0.11	0.85	8	0.34

Figure 1.5 – A tree with different sets of $K = 4$ sensors. The table displays the values of different metrics for the sensor sets. The best values for each metric are in bold type. The remaining possible choices of $K = 4$ sensors within the leaf set are omitted either because they are equivalent to one of the placements considered or because they do not optimize any of the metrics examined.

Hardness of Sensor Placement

Solving sensor placement for trees is significantly easier than in the general case: When the path connecting two any nodes is unique, the number of equivalence classes is tightly linked to the number of nodes lying between two any sensors (see Proposition 1.11). In Chapter 3, we will exploit this fact to derive polynomial-time algorithms in order to optimize sensor placement on trees, with respect to both $\mathcal{P}_s$ and $\mathcal{D}_e$.

For a general network, finding a sensor set $\mathcal{U}$ with $|\mathcal{U}| \leq K$ that optimizes any of the metrics defined in Section 1.2.3 is NP-hard. This can be shown via a reduction to the MDRS problem defined in Section 1.2.2. We prove this result for $\mathcal{P}_s$ [Spinelli et al., 2017a], for the other metrics in Section 1.2.3 it can be derived analogously.

Theorem 1.13. *Given a network $\mathcal{G} = (V, E)$ and a budget K, finding an sensor set $\mathcal{U}$*

which maximizes $\mathcal{P}_s$ is NP-hard.

Proof. We use a reduction from the MDRS, i.e., given a polynomial-time algorithm for solving sensor placement under a budget constraint, we will prove that we can solve the MDRS problem in polynomial time.

Assume that we have a polynomial-time algorithm $\mathcal{A}$ that takes as input a network $\mathcal{G} = (V, E)$ and a budget K, and outputs a set $\mathcal{U} \subseteq V$ of size K such that $\mathcal{P}_s$ is maximized. Recall from Section 1.2.2 that given a network $\mathcal{G}$ and a set $\mathcal{U}$, the probability $\mathcal{P}_s$ can be calculated in time $O(N)$ where $N = |V|$ (it is enough to compute the N distances vector with respect to $\mathcal{U}$ and any reference sensor $u_1 \in \mathcal{U}$). Hence, we will construct an algorithm for the MDRS problem.

Algorithm 1 Solves MDRS given algorithm $\mathcal{A}$ that maximizes $\mathcal{P}_s$ with budget $K \in \mathbb{N}^+$.

```
Require: Network $\mathcal{G} = (V, E)$
  for $K = 1, \dots, |V|$ do
    $\mathcal{U} \triangleq \mathcal{A}(G, K)$
    $P := \mathcal{P}_s(\mathcal{U})$
    if $P = 1$ then
      return $K$
```

Since the full set V always resolves the network, the program is well defined (i.e., it always returns some K). Moreover, it returns precisely the minimum budget K required in order to attain $\mathcal{P}_s = 1$. Last, it is clear that the runtime is at most $O(N(p_{\mathcal{A}}(N) + N))$ where $p_{\mathcal{A}}(N)$ is the runtime of algorithm $\mathcal{A}$. Hence, we would have a polynomial-time algorithm for the MDRS problem, contradicting with this problem being NP-hard. ☐

1.3 Main Related Work

The problem of source localization has been widely studied in recent years and several settings have been considered by different authors. A first survey of the field, focusing especially on the different observation settings, was made available by Jiang et al. [2014]. In this section, we classify the main approaches to the problem, along several relevant axes: epidemic models, observation settings, sensor choices. We also briefly describe important contributions coming from different, but closely-related, fields.

1.3.1 Epidemic Models

Several models of an epidemic spread are studied in the context of source localization. Although discrete-time transmission delays are common [Luo et al., 2014, Prakash et al., 2012, Altarelli et al., 2014a], in order to better approximate realistic settings, many works adopt continuous-time models with varying distributions for the transmission delays; e.g., exponential [Shah and Zaman, 2011, Luo and Tay, 2012] or Gaussian [Pinto et al., 2012, Louni and Subbalakshmi, 2014, Louni et al., 2015, Zhang et al., 2016]. Following this line of work, we also consider transmission delays that follow a continuous distribution, though mainly a distribution with bounded support. For example, we assume truncated Gaussians with which, in contrast to Gaussians, we can ensure positivity (see Section 4.5) or uniform random variables that, among unimodal distribution with fixed support, have the highest variance and give a challenging setting for source localization (see Section 5.2).

Regarding the possible states of each node, many works, including our own, adopt models in which a node can be susceptible (S) or infected (I) and, when it becomes infected, it remains so forever (SI model). Other works, as specified in the discussion below, consider models in which there is a third possible state: After undergoing an infection, a node can become recovered (R) and, in this case, it cannot become infected anymore (SIR model). For the definition of the classic SI and SIR models and a discussion of related research questions, we refer the reader to [Nowzari et al., 2016].

1.3.2 Observation Settings

Complete observation

The first contribution to source localization is by Shah and Zaman [2011]. In this pioneering work, the authors localize the source using a complete observation of the network. They adopt a model in which an infected individual always remains infected (SI) and they assume that, at some unknown time, the state of each node is observed. The source is estimated with a maximum-likelihood (ML) approach that maximizes the probability of the observed infected subnetwork given the source identity. The estimator is derived analytically for regular trees; for general networks, the ML estimator is approximated by assuming that the infection spreads along a breadth-first-search tree (BFS) rooted at the source. Let us call the observed subnetwork of H infected nodes $\mathcal{G}_H$ and let $V_H = \{v_1, .., v_H\}$ be the node set of $\mathcal{G}_H$. A permutation γ of the nodes in V_H is said to be permitted by a source $v \in V_H$ if $\gamma(v_1) = v$ and, starting with the single source $v^\star = v$, it is possible that the infections occur in the order $\gamma(v_1), \gamma(v_2), .., \gamma(v_H)$. Shah and Zaman show that, for regular trees, $\mathbf{P}(\mathcal{G}_H | v^\star = v)$ is proportional to the number of permutations permitted by v, called the *rumor centrality* of v and denoted by $R(\mathcal{G}_H, v)$. Moreover, $R(\mathcal{G}_H, v)$ turns out to be a key quantity also in the approximated computation of the ML estimator for general networks.

Rumor-centrality estimation was later extended to the case of multiple sources [Luo and Tay, 2012] and to the case in which different sources can start spreading the epidemic at different times [Ji and Tay, 2017]. Dong et al. [2013], always within the framework of Shah and Zaman, propose a *local* version of rumor centrality, where an *a priori* knowledge of the set of suspects is used to estimate the source. Wang et al. [2015] study rumor-centrality estimation when multiple independent observations that are originated by a common source are available.

For general networks, Antulov-Fantulin et al. [2014] propose to estimate the source by computing the likelihood of the observed network state. As analytic computation is unfeasible, their method involves large-scale simulations of epidemics starting from all potential sources, which is impractical on large networks, especially when no prior information on the source identity is available.

Prakash et al. [2012] propose to use the minimum-description-length principle [Grünwald, 2007] to identify the set of source nodes by which the infected subnetwork can be most succinctly described. With runtime linear in the number of edges and nodes of the infected subnetwork, their method scales better than rumor-centrality estimation that, even to detect a single source, runs in $O(N^2)$ [Shah and Zaman, 2011]. The results of Prakash et al. [2012] were later extended to the setting where some infected nodes might have been observed erroneously as susceptible: in this case Sundareisan et al. [2015] studied how to determine the number and the identities of both the missing infected nodes and of the sources.

Zhu and Ying [2013] look at a similar setting in which infected nodes can recover (SIR model), but susceptible nodes cannot be distinguished from recovered nodes. They estimate the source by finding the most likely sample path that leads to the observed network-state. The key concept here is the infection eccentricity: the infection eccentricity of a node v is the maximum distance between v and any infected node. Zhu and Ying [2013] show that, for infinite trees, the source associated with the most likely sample path is a node with the minimum infection eccentricity; they call this node a *Jordan infection-center*. The result was extended to general networks and possibility/impossibility results for Jordan-center source localization were derived for Erdős-Rényi networks [Zhu and Ying, 2016].

An algorithm based on dynamic message-passing (DMP) [Kanoria et al., 2011] was proposed by Lokhov et al. [2014]. This algorithm computes the exact likelihood of a node being the source when the network is a tree, and it can be used as approximation for sparse networks. Lokhov et al. [2014] show that DMP estimation outperforms previously proposed methods such as rumor-centrality or Jordan-infection-center estimations on several network topologies. However, the runtime of DMP is is $O(t_0 N^2 \delta)$, where δ is the average degree of the network and t_0 is the time during which the epidemic has spread since its beginning. Furthermore, if t_0 is unknown, the algorithm should be executed for different values of t_0 in order to also optimize over this latter variable.

A practical and intuitive approach, similar in spirit to the Jordan-infection-center estimation, is that of Brockmann and Helbing [2013]: Studying datasets relative to global-scale pandemics (such as the 2009 H1N1 influenza pandemic), they show that, by replacing the mobility-network distance by an *effective distance* that accounts for the fact that some diffusion paths are more likely than others, epidemics spread with homogeneous propagation waves centered at the source. Hence, they suggest to estimate the source by using a snapshot of the network state at a given time to find the node that, taken as center, maximizes the concentricity of the subset of infected nodes.

Partial observation

All previously mentioned works rely on the assumption that, at some point in time, the knowledge of the state of all the nodes is known. This assumption is often not realistic because of both the availability of information and the cost of retrieving it. For this reason, partial observation settings have been studied in the context of source localization. In such settings, only the state of a set of nodes $\mathcal{U}$, which we call *sensors*, is known.

Lokhov et al. [2014] study their DMP method also in the case where only a randomly sampled fraction ξ of the nodes reveal their state. Furthermore, they propose adapted versions of rumor-centrality and Jordan-infection-center for the partial-observation setting. They show that DMP outperforms the other methods also in this setting. However, in order for DMP to identify the source with good accuracy, the fraction ξ has to be very high (in the order of $40 - 60\%$ at least), which is unfortunately unfeasible in many practical situations.

Altarelli et al. [2014a] derived the belief-propagation (BP) equations for the probability distribution of the network state. Their method has the merit of being very general: BP can be used to identify the origin of an epidemic in the SIR, SI, and similar models, even with multiple infection sources and incomplete information. A single iteration of BP can be computed in $O(|E|t_0)$ where $|E|$ is the number of edges and t_0 is the time during which the epidemic has spread since its beginning. However, as remarked by the authors, there are cases where the BP equations do not converge or require too many iterations, especially when the observations available are limited. Their experiments show that the method works well if more than 40% of the nodes are sensors. The BP approach was later extended to the cases in which it is not possible to distinguish between recovered or susceptible nodes and in which the observations are noisy, i.e., they might contain mistakes [Altarelli et al., 2014b].

Zhu and Ying [2014] extend the results of Zhu and Ying [2013] to a partial observation setting in which, at a fixed time, infected nodes reveal their state with a given probability, whereas the state of the rest of the nodes is unknown. Also their method requires around $30 - 40\%$ of the infected nodes to be observed in order to produce an estimated source that is within a few hops from the real source. An extension of this work to multiple sources was also presented in [Zhu et al., 2016].

When the nodes are independently selected to be sensors with node-specific probability q_u, Luo et al. [2014] prove that, for trees, the Jordan-infection-center coincides with the source of the most likely infection path that yields the limited observations available.

The large majority of the approaches discussed above only use information about the *state* of the sensors and do not assume that the *infection times* of the sensors are known. This is, in many cases, an unnecessary limitation because, by interviewing users of a social network or patients affected by a disease, a (possibly noisy) observation of the infection time might become available [Zhu et al., 2015]. Moreover, infection times are truly an asset in source localization: Indeed, if we observe the infection time of sufficiently many sensors and the node-to-node transmission delays are known, the source can always be correctly identified [Chen et al., 2014], whereas otherwise this is not true.

In this direction, Pinto et al. [2012] introduce a model where the source is estimated based on the infection times of a small set of sensors that are *a priori* chosen in the network. They also make the additional assumption that the infected sensors can reveal by which neighbor they were infected. Assuming that the transmission delays are Gaussian random variables, a ML estimator of the source, given the available observations, can be computed for trees in $O(N)$. For general networks, an approximate ML estimator can be computed by using a standard BFS approximation, in $O(N^3)$. From an application point of view, an interesting feature of this model is the *a priori* choice of the sensors: it can be used for all the problems in which knowing the state of the nodes might have a nontrivial cost and the budget for sensors is limited. Another model for source estimation based on *a priori* chosen sensors is proposed by Seo et al. [2012]. However, differently from [Pinto et al., 2012], this work defines a ranking of the most likely sources by using only the information about the sensor state. Instead, another work that uses the observation of the infection times of a subset of the nodes is that of Zhu et al. [2015]. They propose to estimate the source computing the most likely spreading tree that is consistent with the observed infection times. This approach is shown to outperform that of Pinto et al. [2012] when the fraction of the observed infection times is large enough (larger than 20% for most of the experiments presented by Zhu et al. [2015]).

A different sparse-observation setting was explored by Kumar et al. [2017] who estimate the source using either a list of infector-infected nodes or a list of node pairs where it is known which node of each pair was infected first (and the two nodes in the pairs are not necessarily neighbors). This framework is interesting because it is extremely light in terms of assumptions: the knowledge of the infection times, and even of the diffusion model, is not required. The source is detected with a Markov-chain-Monte-Carlo scheme whose performance is highly sensitive to the fraction of information available: The experiments show that to guarantee a high detection-rate, the infection precedence has to be revealed for a large fraction of the node pairs (more than 60% for most of the networks examined).

Finally, Farajtabar et al. [2015] look at the case in which several sparse epidemic traces are available. They tackle the more general problem of both learning the contact network (using multiple cascades) and the source identity (using one or more cascades generated by the same source). Their approach is based on an efficient importance-sampling approximation of the likelihood and was successfully applied to identify the origin of memes in social networks.

1.3.3 Sensor Placement

The problem of finding the *minimum* number of sensors required to correctly identify the source is studied, for deterministic transmission delays, by Zejnilović et al. [2013] who solve it on trees under the assumption that the time at which the epidemic starts is known. In this setting, minimizing the number of sensors needed for source localization is equivalent to finding a minimum-size resolving set (RS) of the network [Chartrand et al., 2000]. Without assuming a tree topology and a known starting time, approximation algorithms for the same problem were developed by Chen et al. [2014], still with deterministic transmission delays, using the connection to the problem of finding a double resolving set (DRS) of a network [Cáceres et al., 2007]. In fact, DRS are the natural counterpart of RS for the case in which the starting time of the epidemic is unknown. The results of [Chen et al., 2014] were also extended to the case of multiple sources by Zhang et al. [2015]. However, these results are not very practical: In a network of size N, the minimum number of sensors required can be up to $N-1$ (e.g., for a complete network). This makes the application of this approach difficult in most real-world situations where sensor choices cannot be solved without taking budget limitations into account. For this reason, the problem of choosing a fixed number of sensors is of crucial importance.

Pinto et al. [2012] evaluate the effect of choosing sensors according to several natural heuristics, e.g., using high-degree nodes or optimizing for distance centrality. Later, Louni and Subbalakshmi [2014] and Louni et al. [2015] propose, for a similar model, to place the sensors using a betweenness-centrality criterion [Freeman, 1977]. These and other heuristic approaches for sensor placement are also evaluated empirically by Seo et al. [2012] who reach the conclusion that, among the placements they consider, the betweenness-centrality criterion performs the best. In a more recent work, Zhang et al. [2016] introduce a new heuristic for the choice of sensors, called *coverage-rate*. This is linked to the total number of nodes that are neighbors of the sensors. The authors show that an approximated optimization of this metric outperforms several heuristics, including betweenness-centrality, highest-degree and closeness-centrality. We remark here that none of the above-mentioned heuristics is *directly* linked to source localization.

When the variance of the transmission delays is very small, Zejnilović et al. [2015b] compute the optimal sensor placement based on the analysis of error exponents. This is an interesting approach with strong mathematical foundations. Unfortunately, it is applicable only to small networks because it requires computing the error exponent for every possible sensor set. Moreover, it requires the error probability in source localization to be vanishing, i.e., sufficiently many sensors must be available.

In [Celis, Pavetić, Spinelli, and Thiran, 2015] we present a polynomial-time algorithm for optimally placing a fixed number of sensors in a tree where the transmission delays have bounded noise. This contribution is based on dynamic-programming techniques and, as it heavily relies on the tree structure of the network, it is not directly extendible to different topologies; see Chapter 3.

In [Spinelli et al., 2016, 2017a], we propose algorithms for choosing sensors in general networks with an approach that is specifically tailored to source localization. In particular, we present methods that use the connection between sensor placement and the double-resolving-set problem [Cáceres et al., 2007]. These methods give favorable results when compared with both betweenness centrality and coverage-rate. In contrast with previous approaches, we also account for the variance in the transmission delays and show that the optimal sensor choice depends on the transmission variance; see Chapter 4.

1.3.4 Online Sensor Placement

Online resource allocation is studied in several contexts, including information diffusion [Dhamal et al., 2016], curing policies for epidemics [Drakopoulos et al., 2014, Scaman et al., 2016] and more general *robust-optimization* problems where, to reach some objective, only a part of the resources is allocated *a priori* and the rest is deployed, possibly at a higher cost, when more information is available [Gupta et al., 2010]. Another related line of work in the field of artificial intelligence is that of *active learning*, the study of how to adaptively take a sequence of decisions by using sparse data, in order to optimize a given objective [Settles, 2012, Golovin and Krause, 2011].

In the context of source localization, Zejnilović et al. [2015a] propose an algorithm that sequentially places sensors to localize the source *after* the epidemic has spread through the entire network. Working under the hypothesis that the starting time of the epidemic is known and that the transmission delays are deterministic, they give a dynamic-programming algorithm for localizing the source by using the minimum possible number of sensors. In order to reduce the computational costs, they also propose a greedy approximation for which guarantees are proven using adaptive submodularity [Golovin and Krause, 2011].

Adopting different techniques, in [Spinelli et al., 2017b, **?**], we propose a general algorithm for source localization and sensor placement that allocates sensors one after the other and localizes the source by iteratively refining a set of candidate sources. This approach has two important features: The sensors can be placed *while* the epidemic evolves and the transmission delays can have high variance; see Chapter 5.

1.3.5 Other Related Work

Localization problems are also studied in very different settings such as the estimation of the source of a signal [Li et al., 2016], of heat sources [Masood and Zaman, 2004] or of pollution sources [El Badia and Ha-Duong, 2002]. In all these inverse problems, a diffusion takes place in a continuous medium and the techniques used are, in general, very different from those used for source localization on networks. However, Pena et al. [2016] adapted the continuous setting of Candès and Fernandez-Granda [2014] to network source localization under the hypothesis that the epidemic occurs in a way that mimics heat diffusion in a continuous medium.

In the network domain, a closely related problem to source localization is that of *outbreak detection*, i.e., detecting the existence of an epidemic in a timely manner. In this context, sensor placement is a well-studied problem. The optimal solution for timely detection is to place sensors at the *k-Medians* [Berry et al., 2006]. Furthermore, the optimization of several alternate metrics of interest (e.g., the percentage of infected population at the time of detection) is studied by Leskovec et al. [2007b] and Krause et al. [2008].

A problem that can, in a sense, be considered the dual of source localization is that of *source obfuscation*. Here the interest is in hiding the source of a diffusion, i.e., in spreading a message anonymously. Anonymous spread is important, for example, in contexts where there is no freedom of expression, hence authors of sensitive messages want to remain anonymous. Fanti et al. [2015] propose a randomized diffusion-protocol that spreads content fast and, on regular trees, achieves perfect obfuscation of the source. In the same line of work, Venkatakrishnan et al. [2017] study a way to redesign the Bitcoin network and its broadcasting protocol in order to guarantee transaction anonymity. Luo et al. [2016], instead, model diffusion and source localization as a strategic game, where the source and the network administrator are the two players; they study the best-response infection strategy for the source and the best-response estimation strategy for the administrator. In all these works, the main difference with the traditional source localization setting is the presence of a powerful network administrator who can act on the propagation dynamics by purposely delaying some infections.

Finally, *viral marketing* [Richardson and Domingos, 2002, Kempe et al., 2003, Leskovec et al., 2007a] has many similarities with sensor placement. Whereas in sensor placement we want to identify the nodes that optimize information collection, in viral marketing we want to find the nodes that optimize information diffusion. With an analysis based on submodularity properties, Kempe et al. [2003] propose approximation algorithms for finding the most influential nodes in a large class of diffusion models. In spite of the similar formulation of the two problems, however, similar techniques cannot be used to optimize sensor placement for source localization (see Sections 4.3.2 and 4.4.2).

2 Double Metric Dimension (DMD) of Random Networks

In Section 1.2.2, we defined the double metric dimension (DMD) of a network and we explained its relevance to source localization. In a setting where the infection times of some special nodes —called sensors— are observed, where the starting time of epidemics is unknown and where epidemics propagate with deterministic transmission delays, the source is always correctly and univocally identified if and only if the sensor set is a double resolving set (DRS) of the network. In other words, the minimal number of sensors required to guarantee source localization is the DMD of the network.

Therefore, the DMD of a network is an important metric for quantifying how difficult it is to localize the source in a particular network. Let us consider some examples. In a star network (see Figure 2.1a) the DMD is $N-1$: a set that does not contain all the external nodes cannot be a double resolving set. Whereas, for a path network, instead, the DMD is 2 (see Figure 2.1b): taking as sensors the two nodes at the extremities of the path suffices to always identify the source correctly (this is a particular case of the result proven for trees in Proposition 1.6). These examples show that it is much easier to detect the source of an epidemic in a path-like network than in a star-like network.

Unfortunately, computing the DMD of general networks is NP-hard [Kratica et al., 2009]. Chen et al. [2014] showed that there does not exist any $(1-\varepsilon)\log N$-approximation

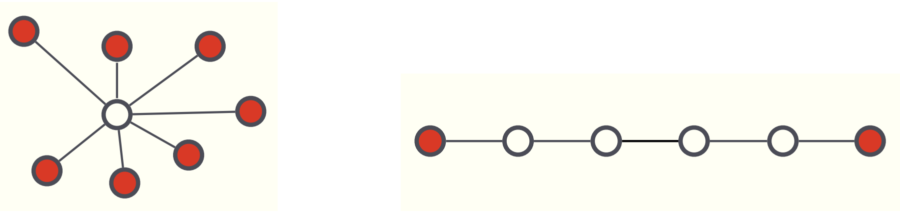

(a)
(b)

Figure 2.1 – **(a)**: The DMD of a star network with N nodes is equal to $N-1$: every double-resolving set contains all the leaves. **(b)**: The DMD of a path network is equal to 2: the two extreme nodes form a minimal-size double-resolving set.

algorithm for DMD for any $\varepsilon > 0$[1] and gave a $(1 + o(1)) \log N$-approximation algorithm. In this chapter, we take a complementary approach to the approximation of the DMD, i.e., we derive asymptotic bounds. Our main result is an upper bound for the DMD of $\mathcal{G}(N, p)$ random networks [Erdős and Rényi, 1959]. This bound undergoes a non-monotonic behavior in the parameter p. Through an experimental analysis we show that, indeed, the DMD of $\mathcal{G}(N, p)$, random networks is non-monotonic in p and that we can identify parameter regimes where source localization is substantially more difficult than in other regimes.

A problem that is closely related to the determination of the DMD is the determination of the metric dimension (MD) of a network. With respect to source localization, the MD is the minimum number of sensors needed to localize the source when the additional information of the starting time of the epidemic is available. Results that are similar to those presented in this chapter were derived for the MD problem by Bollobás et al. [2013]. Our proof techniques are inspired by theirs; however, as we explain in Section 2.2.2, bounding the DMD is more challenging than bounding the MD. Along with the derivation of our results, we will highlight where and why the specificity of our problem requires a different approach.

2.1 Overview

In Section 2.2, we give some introductory elements, including a definition of the MD, its relationship to the DMD, and a short survey of the results available for the MD problem.

In Section 2.3, we derive an asymptotic upper bound for the DMD of $\mathcal{G}(N, p)$ (or Erdős-Rényi) random networks.

We take $p = p(N)$ such that the $\mathcal{G}(N, p)$ networks are dense enough (in a sense that will be specified in Section 2.3) and we give an asymptotic upper bound for three different regimes of $p(N)$: when $p(N) = \Theta(1)$, when $p(N) \sim N^{-i/(i+1)}$ for some $i \in \mathbb{N}^+$ and, finally, when $p(N) = o(1)$ but $p(N)$ is not asymptotic to $N^{-i/(i+1)}$ for any $i \in \mathbb{N}^+$. In the regimes $p(N) = \Theta(1)$ and $p(N) \sim N^{-i/(i+1)}$, the bound reaches its local minima.

In Section 2.4, we experimentally show that the DMD approximation obtained through the greedy algorithm of Chen et al. [2014], and even the probability of error of source localization when using a limited number of sensors, closely follow our theoretical bound. We also observe that, on $\mathcal{G}(N, p)$ networks, the reults obtained via a greedy approximation of the DMD resembles very closely to those obtained via a greedy approximation for the MD, suggesting that, for a given accuracy, ignoring the starting time of the epidemic does not incur a much higher cost in terms of sensors. We also observe that this is not

---

[1] Unless NP $\subset$ DTIME($N^{\log \log N}$). See [Garey and Johnson, 2002] for a definition and discussion of complexity classes.

the case for other topologies, e.g., trees.

2.2 Preliminaries

2.2.1 The Metric Dimension (MD)

The metric dimension of a network $\mathcal{G}(V, E)$ is the minimum number of nodes in a subset $\mathcal{U}$ of the node set V such that all other nodes are uniquely determined by their distances to the nodes in $\mathcal{U}$. More formally, we have the following definitions.

Definition 2.1 (Resolving set (RS))**.** *Let $\mathcal{G}(V, E)$ be a network and $\mathcal{U} \subseteq V$. $\mathcal{U}$ is a resolving set of $\mathcal{G}$ if and only if, for every $v, w \in V$ with $v \neq w$, there exists $u \in \mathcal{U}$ such that $d(v, u) \neq d(w, u)$.*

Definition 2.2 (Metric dimension (MD))**.** *Let $\mathcal{G}(V, E)$ be a network. The metric dimension of $\mathcal{G}$ is the minimum $k \in \mathbb{N}^+$ such that there exists a resolving set $\mathcal{U}$ of $\mathcal{G}$ with $|\mathcal{U}| = k$.*

Figure 2.2 shows resolving sets for the two examples of Figure 2.1.

The problem of determining the metric dimension of a network is often called the metric-dimension problem. Denoting by $\mathrm{MD}(\mathcal{G})$ the metric dimension of a network $\mathcal{G}$, we have that $1 \leq \mathrm{MD}(\mathcal{G}) \leq N - 1$. Moreover $\mathrm{MD}(\mathcal{G}) = 1$ if and only if $\mathcal{G}$ is a path and $\mathrm{MD}(\mathcal{G}) = N - 1$ if and only if $\mathcal{G}$ is the complete network [Chartrand et al., 2000].

The MD was first defined by Slater [1975] and, independently, by Harary and Melter [1976]; however Erdős and Rényi studied (and solved asymptotically) the metric dimension problem in the special case of hypercubes earlier [Erdős and Rényi, 1963]. Resolving sets are studied in various application areas including drug discovery [Chartrand et al., 2000], robot navigation [Khuller et al., 1996], and network discovery and verification [Beerliova et al., 2006].

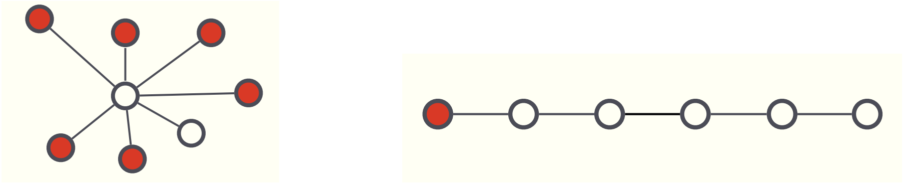
In the context of source localization, the MD of a network is the minimum number of sensors required to guarantee correct source identification when the transmission delays are deterministic and the starting time of epidemics is known. This model is considered by Zejnilović et al. [2013] who show the linked between optimal sensor placement and resolving sets.

Solving the metric-dimension problem for an arbitrary network is an NP-complete problem [Garey and Johnson, 1979, Khuller et al., 1996]. Hauptmann et al. [2012] show that the MD of an arbitrary network cannot be approximated within a factor of $(1 - \varepsilon) \log(N)$ for any constant $\varepsilon > 0$[2] and gave a $2 \log N$-approximation algorithm.[3] Furthermore,

---

[2] Unless $\mathrm{NP} \subset \mathrm{DTIME}(N^{\log(\log N)})$.

[3] More exactly, their algorithm has approximation ratio $1 + \log N + o(\log N)$.


(a)

(b)

Figure 2.2 – **(a)**: The MD of a star network with N nodes is equal to $N-2$: Every RS contains at least all but one of the leaves. **(b)**: The MD of a path network is equal to 1: one extreme node forms a minimal-size RS.

researchers have determined metric dimensions for many particular classes of networks, such as trees [Chartrand et al., 2000, Harary and Melter, 1976, Khuller et al., 1996] (see also Proposition 2.6 here below), cycles [Chartrand et al., 2000] and wheels [Shanmukha et al., 2002].

For the reader's convenience we recall the definition of DMD and DRS.

Definition 2.3 (Double resolving set (DRS)). *Let $\mathcal{G}(V,E)$ be a network and $\mathcal{U} \subseteq V$ with $|\mathcal{U}| \geq 1$. $\mathcal{U}$ is a double resolving set of $\mathcal{G}$ if and only if, for every $v, w \in V$ with $v \neq w$, there exist $u_1, u_2 \in \mathcal{U}$ such that $d(v,u_1) - d(v,u_2) \neq d(w,u_1) - d(w,u_2)$.*

Definition 2.4 (Double metric dimension (DMD)). *Let $\mathcal{G}(V,E)$ be a network. The double metric dimension of $\mathcal{G}$ is the minimum $k \in \mathbb{N}^+$ such that there exists a double resolving set S of $\mathcal{G}$ with $|\mathcal{U}| = k$.*

Looking at Figures 2.1 and 2.2, we see that in both cases the MD of the network is smaller than the DMD. As we formalize in the following lemma, this is true in general.

Lemma 2.5. *Let $\mathcal{G}(V,E)$ be a network.* $\mathrm{MD}(\mathcal{G}) \leq \mathrm{DMD}(\mathcal{G})$.

Proof. Let $\mathcal{U} \subseteq V$ be a double resolving set of $\mathcal{G}$. We show that $\mathcal{U}$ is also a resolving set of $\mathcal{G}$, from which we can deduce that $\mathrm{MD}(\mathcal{G}) \leq \mathrm{DMD}(\mathcal{G})$. Let $v, w \in V$ with $v \neq w$. Since $\mathcal{U}$ is a double resolving set of $\mathcal{G}$, there exist $u_1, u_2 \in \mathcal{U}$ such that $d(v,u_1) - d(w,u_1) \neq d(v,u_2) - d(w,u_2)$. Hence, at least one between $d(v,u_1) - d(w,u_1)$ and $d(v,u_2) - d(w,u_2)$ is different from zero. We conclude that for every $v, w \in V$ there exist $s \in \mathcal{U}$ such that $d(v,s) \neq d(w,s)$, hence $\mathcal{U}$ is a resolving set of $\mathcal{G}$. $\square$

Note that the strict inequality $\mathrm{MD}(\mathcal{G}) < \mathrm{DMD}(\mathcal{G})$ does not hold in general: For a complete network, $MD(\mathcal{G}) = DMD(\mathcal{G}) = N-1$.

In Section 1.2, we proved that the DMD of a tree is equal to the number of its leaves (see Proposition 1.6) and that any double resolving set must contain all leaves. Instead, the metric dimension of a tree is strictly smaller than the number of its leaves. In particular,

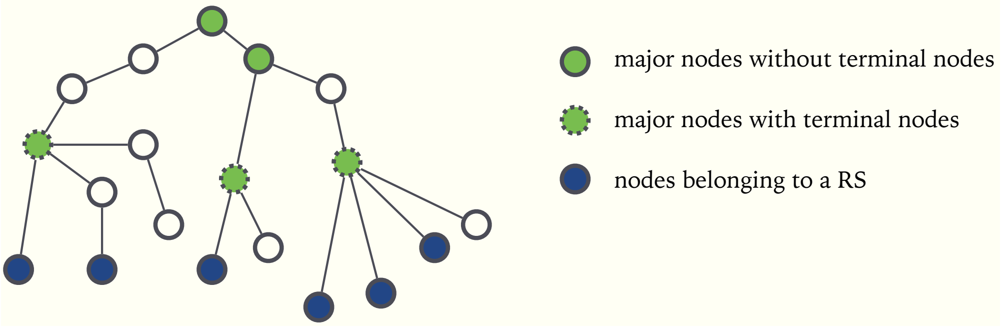

major nodes without terminal nodes
major nodes with terminal nodes
nodes belonging to a RS

Figure 2.3 – In a tree, it is possible to find a minimum-size RS $\mathcal{U}$ by including in $\mathcal{U}$, for every major node, all its terminal nodes except one (see Proposition 2.6).

the following result, proved, e.g., by Chartrand et al. [2000], and illustrated in Figure 2.3 holds.

Proposition 2.6 (MD of trees)**.** *Let* $\mathcal{T} = \mathcal{T}(V, E)$ *be a tree that is not a path and let* $\mathcal{L}$ *be the set of all leaves of* $\mathcal{T}$*. For every node* $u \in V$*, we say that* u *is a major node of* $\mathcal{T}$ *if its degree is strictly larger than* 2*. Let* $u \in V$ *be a major node and* $\ell \in \mathcal{L}$ *be a leaf of* $\mathcal{T}$*. We say that* ℓ *is a terminal node of* u *if, for every other major node* $v \in V$, $d(u, \ell) < d(v, \ell)$*. Call* $\mathrm{ex}(\mathcal{T})$ *the number of major nodes of* $\mathcal{T}$ *that have at least one terminal node. Then,*

$$\mathrm{MD}(\mathcal{T}) = |\mathcal{L}| - \mathrm{ex}(\mathcal{T}).$$

In particular, we can construct a minimum-size RS $\mathcal{U}$ of a tree $\mathcal{T}$ by including in $\mathcal{U}$, for any major node of $\mathcal{T}$, all its terminal nodes except one. In Section 2.4.2, we will present experimental results concerning the MD of trees in comparison with their DMD.

The MD of Random Networks

The MD of Erdős-Rényi networks was studied in detail in a paper by Bollobás et al. [2013]. Bounding the MD from above and from below, they show that the MD follows a *zigzag* behavior as a function of p. The intuition behind this result, as stated in their paper, is as follows:

If a random graph is sufficiently dense, then the graph locally (that is, "observed" from a given vertex) "looks" the same. In other words, the cardinality of the set of vertices at a certain graph distance from a given vertex v *does not differ much for various* v*. After grouping the vertices according to their graph distances from* v*, it turns out that for the metric dimension the ratio between the sizes of the two largest groups of vertices is of crucial importance. If these two groups are roughly of the same size, then a typical vertex added to the resolving set distinguishes a lot of pairs of vertices, and hence the metric dimension is small. If, on the other hand, these two groups are very different in size, a*

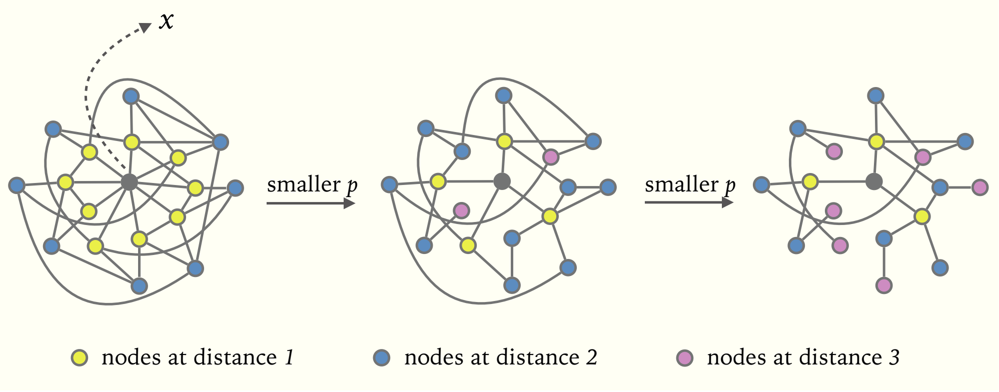
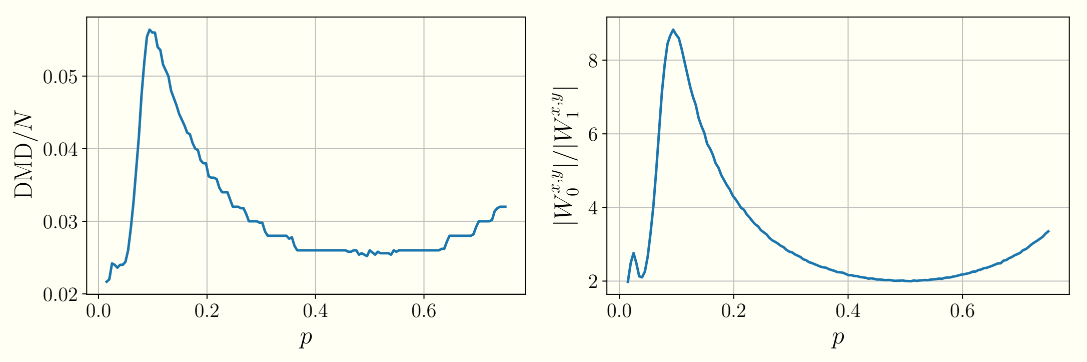


Figure 2.4 – The MD is of $\mathcal{G}(N, p)$ networks is non-monotone in p. When p is large (left), if we add a sensor in x, we can distinguish many pairs of nodes (any node at distance 1 from x and any node at distance 2 from x); when p gets smaller (center), some nodes at distance 3 from x appear and x distinguishes a smaller number of pairs because there is a large group of nodes at distance 2; when p gets even smaller (right), the cardinality of the nodes at distance 3 from x becomes similar to the cardinality of the nodes at distance 2 from x and the number of pairs that can be distinguished by x increases again. The same argument holds, when p becomes even smaller, for groups of nodes at larger distances from x.

typical vertex distinguishes those few vertices belonging to the second largest group from the rest. The number of other pairs that are distinguished is negligible and hence the metric dimension is large. It is clear that this parameter is non-monotonic. Let us start with a random graph with constant edge probability p. For each vertex v in the graph, a constant fraction of all vertices are neighbours of v and a constant fraction of vertices are non-neighbours. When decreasing p, the number of neighbours decreases, and some vertices will appear at graph distance 3. As a result, the metric dimension increases. Continuing this process, the number of vertices at graph distance 3 increases more and more, and at some point this number is comparable to the number of vertices at graph distance 2. Then, the metric dimension is small again, and the same phenomenon appears in the next iterations.

An illustration of this argument is provided by Figure 2.4. We will see in Section **??** that the zigzag behavior of the MD is not peculiar to $\mathcal{G}(N, p)$ networks: a similar non-monotonicity of the MD (and the DMD) with respect to the network density can be observed also in preferential attachment models.



(a) (b)

Figure 2.5 – $\mathcal{G}(N, p)$ networks with $N = 500$. **(a)**: Approximation of the DMD obtained via the approximating algorithm of Chen et al. [2014] . **(b)**: Average ratio between the number of nodes at equal distance from a pair of nodes ($|W_0^{x,y}|$) and the number of nodes at relative distance 1 from a pair of nodes ($|W_1^{x,y}|$).

2.2.2 Contrast between the MD & the DMD

For the DMD, the equivalent of the set of nodes at a given distance k from a node x (see Section 2.2.1) are the set of nodes at *relative* distance k from a pair of nodes x and y, i.e.,

$$W_k^{x,y} \triangleq \{v \in V : d(v, x) - d(v, y) = k\}.$$

Figure 2.5a and Figure 2.5b show, respectively, the average ratio between $|W_0^{x,y}|$ and $|W_1^{x,y}|$ as a function of p (the average is taken over all possible pairs (x, y)) and the approximation of the DMD obtained with the greedy algorithm by Chen et al. [2014]. The two curves follow closely the same pattern, hinting to the close relationship between the relative size of the sets $\{W_k^{x,y}\}_{k\in\mathbb{N}}$ and the DMD. This suggests that a result similar to the one derived by Bollobás et al. [2013] for the MD can be derived for the DMD.

In Lemma 2.5, we proved that for any network $\mathcal{G}$, $\mathrm{MD}(\mathcal{G}) \leq \mathrm{DMD}(G)$. Consequently a lower bound for the MD is a lower bound on the DMD, and to obtain a lower bound for the DMD, we can directly apply the results of Bollobás et al. [2013]. For this reason, we focus on the derivation of upper bounds for the DMD.

Compared to the MD, bounding the DMD is technically more challenging. If we want to prove that a set $\mathcal{U}$ is a RS for a network $\mathcal{G}(V, E)$ it suffices to show, for any $x, y \in V$ with $x \neq y$, that at least one node in $\mathcal{U}$ has a different distance from x than from y. Whereas, if we want to prove that $\mathcal{U}$ is a DRS, we must show, for any $x, y \in V$ with $x \neq y$ that there exist two nodes u_1, u_2 such that

$$d(u_1, x) - d(u_2, x) \neq d(u_1, y) - d(u_2, y). \tag{2.1}$$

Proof steps	*Comparison with the MD case*
1. For two arbitrary nodes x, y, define two sets A_{xy} and A_{yx} such that any pair $(v, w) \in A_{xy} \times A_{yx}$ distinguishes x and y.	For MD, it suffices to define one set A_{xy} such that any node $v \in A_{xy}$ distinguishes x and y.
2. Lower bound the cardinality of A_{xy} and A_{yx}.	
3. Upper bound the probability that a set of size w sampled uniformly at random contains at least one pair in $A_{xy} \times A_{yx}$.	For MD, it suffices to upper bound the probability that the random set contains at least one node in A_{xy}.
4. Use (**3**) to derive an upper bound on the minimal size of a DRS.	

Table 2.1 – Sketch of the proofs

Note that, (2.1) can hold even if $d(u_1, x) = d(u_1, y)$ or $d(u_2, x) = d(u_2, y)$, and that, it is not sufficient that $d(u_1, x) \neq d(u_1, y)$ and $d(u_2, x) \neq d(u_2, y)$ for (2.1) to hold. Therefore, as we will see in the proof of our theorems, the study of DMD requires a more fine-grained analysis of the network topology than the study of the MD.

Table 2.1 summarizes the steps we take to derive upper bounds for the DMD (see Theorems 2.10, 2.11 and 2.21). We underline the main differences with respect to the derivation of upper bounds for the MD as it is done by Bollobás et al. [2013].

2.2.3 Definitions & Useful Lemmas

We study the DMD for $\mathcal{G}(N, p)$ random networks [Erdős and Rényi, 1959] where N denotes the number of nodes and p is the probability that every given edge appears in the network. A network $\mathcal{G}$ can be sampled from the $\mathcal{G}(N, p)$ model by sampling $\binom{N}{2}$ independent Bernoulli random variables $\{Z_{uv}\}$, one for each unordered pair of nodes, with success probability p: if $Z_{uv} = 1$, then $uv \in E$, otherwise $uv \notin E$.

In this chapter, we assume that all edges have the same weight, i.e., $w_{uv} = 1$ for every $uv \in E$, hence the distance $d(x, y)$ between two nodes x and y is equal to the minimum number of edges on a path connecting x and y.

We denote by δ the expected average degree of every node, i.e., $\delta \triangleq p(N-1)$.

In the following, all asymptotics are as $N \to +\infty$. Moreover, we say that a statement $\mathcal{E}$

holds asymptotically almost surely (a.a.s.) if and only if $\mathbf{P}(\mathcal{E}) \to 1$ as $N \to +\infty$.

Let $f, g : \mathbb{N} \to \mathbb{R}$ be two functions. If $\exists\ C_1, C_2 \in \mathbb{R}$ such that, for large enough n, $C_1 f(n) \leq g(n) \leq C_2 f(n)$, we say $f = \Theta(g)$ and we sometimes write $f \sim g$. Moreover we use the standard notations $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$ for $n \to +\infty$ and $f(n) = \omega(g(n))$ if $f(n)/g(n) \to +\infty$ for $n \to +\infty$.

We will often look at the set of nodes at some distance $i \in \mathbb{N}$ from a given node. Hence we will use the following definitions.

Definition 2.7. *Let $\mathcal{G}(V, E)$ be a network, $v \in V$ and $i \in \mathbb{N}^+$. We denote the set of nodes at distance i from v by*

$$S(v, i) \triangleq \{u \in V : d(u, v) = i\}.$$

Moreover, the set of nodes at distance at most i from v will be denoted by

$$N(v, i) \triangleq \{u \in V : d(u, v) \leq i\}.$$

We will also sometimes consider the sets $S(Z, i)$ and $N(Z, i)$ for $Z \subseteq V$ where the definitions of $S(Z, i)$ and $N(Z, i)$ follow from Definition 2.7 in a natural way. We will estimate the cardinality of some relevant subsets of V using the following lemma; it is a consequence of the Chernoff bound for the case of binomial random variables.

Lemma 2.8 (Chernoff bound for binomial random variables. Corollary 2.3 in Janson et al. [2011])**.** *Let X be a binomial random variable and $0 < \varepsilon < \frac{3}{2}$*

$$\mathbf{P}(|X - \mathbf{E}[X]| \geq \varepsilon \mathbf{E}[X]) \leq 2 \exp\left(- \frac{\varepsilon^2 \mathbf{E}[X]}{3} \right).$$

Moreover, we will prove our bounds by using the probabilistic method, as formulated in the next lemma.

Lemma 2.9 (Probabilistic method)**.** *Let $\mathcal{G}(V, E)$ be a network. Let $w \geq 2$ and let W be a set sampled uniformly at random from the set collection $\{Z \subseteq V : |Z| = w\}$. If, for all $x, y \in V$ with $x \neq y$ the probability that W does not distinguish x, y is at most $1/N^2$, then $\mathrm{DMD}(\mathcal{G}) \leq w$.*

Proof. Let us denote by X the number of node pairs that are not distinguished by W. Assume by contradiction that $\mathrm{DMD}(\mathcal{G}) > w$, which implies that no set of size w is a DRS, hence that $X \geq 1$. We have that

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{u \neq v} \mathbf{P}(\{(u,v) \text{ is not distinguished by } W\}) \\
&\leq \sum_{u \neq v} \frac{1}{N^2} \\
&\leq \frac{N^2}{2} \cdot \frac{1}{N^2} = \frac{1}{2},
\end{aligned}$$

which gives a contradiction with $X \geq 1$. □

2.3 Upper Bounds on the DMD

In this section, we derive an upper bound on the DMD of $\mathcal{G}(N,p)$ random networks for a wide range of the parameter p.

Table 2.2 summarizes our main results and points to the specific theorems that prove them. For all these results, our proofs follow the structure of the first column of Table 2.1: first, we define, for every $x, y \in V$ some *distinguishing sets* A_{xy} and A_{yx}, i.e., some sets A_{xy} and A_{yx} such that any pair $(u,v) \in A_{xy} \times A_{yx}$ distinguishes x and y; second, we lower bound the cardinality of the distinguishing sets; finally, we use this lower bound to derive an upper bound on the DMD. In each of the three different regimes of p we consider, the main challenge is to define appropriate distinguishing sets whose cardinality is large enough.

In Theorem 2.10 (Section 2.3.1), we start with the case $p = \Theta(1)$ in which it is easy to find distinguishing sets whose size is a constant fraction of N. Next, in Section 2.3.2, we move to the case $p = o(1)$ where we study two different regimes for p. First, we consider $p = \Theta(N^{-i/(i+1)})$ for some $i \in \mathbb{N}^+$ (Theorem 2.11). In this case, with a slightly more complicated derivation with respect to Theorem 2.10, we can still find distinguishing sets whose size is a constant fraction of N. Second, we study the intermediate regimes, i.e., those in which $p \neq \Theta(N^{-i/(i+1)})$ for all $i \in \mathbb{N}^+$ but there exists $i \in \mathbb{N}$ such that $p = o(N^{-i/(i+1)})$ and $p = \omega(N^{-(i+1)/(i+2)})$. In this last regime, identifying distinguishing sets which are large enough is more difficult and, whereas in the two previous cases we obtained an upper bound of order $\log(N)$, the upper bound is here of order $\omega(\log(N))$.

Note that, with these three results we cover all values of p such that $p = \Theta(1)$ or such that $p = o(1)$ and $p = \Omega(1/N^{1-x})$ for some $x > 0$.

In Section 2.3.3, we summarize all our results (Theorem 2.22) and compare them with the results obtained by Bollobás et al. [2013] for the MD.

Values of p	*Order of the upper bound*	*Statement*
$p = \Theta(1)$	$\log(N)$	Theorem 2.10
$\exists\ i \in \mathbb{N}^+$: $p = \Theta(N^{-i/(i+1)})$	$\log(N)$	Theorem 2.11
$\exists\ i \in \mathbb{N}$: $p = o(N^{-i/(i+1)})$ and $p = \omega(N^{-(i+1)/(i+2)})$ (intermediate regimes)	$\omega(\log(N))$	Theorem 2.21

Table 2.2 – Upper bounds for the DMD of $\mathcal{G}(N,p)$ networks.

2.3.1 Bound for $\mathcal{G}(N,p)$ with $p = \Theta(1)$

Theorem 2.10. *Let* $p = \Theta(1)$ *and let* $z \triangleq 1 - p(1-p) = p^2 + (1-p)$*. Let* $\mathcal{G} \in \mathcal{G}(N,p)$*. Then a.a.s.*

$$\mathrm{DMD}(\mathcal{G}) \leq \frac{2\log(\sqrt{2}N)}{\log(1/z)}. \tag{2.2}$$

Proof. We prove this theorem by applying Lemma 2.9: We upper bound the probability that a set W of cardinality w picked uniformly at random does not distinguish a pair $x, y \in V$ and show that, for our upper bound to be smaller than $1/N^2$, it suffices that

$$w \leq \frac{2\log(\sqrt{2}N)}{\log(1/z)}.$$

For $u \neq v$, we define the set of the nodes that are neighbors of u but have distance greater or equal than 2 from v as

$$A_{uv} \triangleq S(u,1) \backslash N(v,1) = \{z : d(u,z) = 1, d(v,z) \geq 2\} \tag{2.3}$$

Let $x \neq y$. A_{xy} and A_{yx} are distinguishing sets for x and y, i.e., for every $u \in A_{xy}$ and $v \in A_{yx}$, x and y are distinguished by (u,v). This holds because

$$d(x,u) - d(x,v) = 1 - d(x,v) \leq -1$$

and

$$d(y,u) - d(y,v) = d(y,u) - 1 \geq 1.$$

Let W be a set picked uniformly at random amongst all the subsets of V of size w and call p_w the probability that W does not distinguish x and y. We have

$$\begin{aligned}
p_w &\leq \mathbf{P}(\nexists u, v \in W : (u,v) \in A_{xy} \times A_{yx}) \\
&\leq \mathbf{P}(\{\nexists u \in W : u \in A_{xy}\} \cup \{\nexists v \in W : v \in A_{yx}\}) \\
&= \mathbf{P}(\{W \subseteq V \backslash A_{xy}\} \cup \{W \subseteq V \backslash A_{yx}\}) \\
&\leq \mathbf{P}(W \subseteq V \backslash A_{xy}) + \mathbf{P}(W \subseteq V \backslash A_{yx}) \\
&\leq 2\mathbf{P}(W \subseteq V \backslash A_{xy}) \\
&\leq 2\left(\frac{N - |A_{xy}|}{N}\right)\left(\frac{N-1-|A_{xy}|}{N-1}\right)\cdots\left(\frac{N-w+1-|A_{xy}|}{N-w+1}\right) \\
&\leq 2\left(1 - \frac{|A_{xy}|}{N}\right)^w. \qquad (2.4)
\end{aligned}$$

Now we estimate $|A_{xy}|$. First, we have

$$\mathbf{E}[|A_{xy}|] = (N-2)p(1-p).$$

Second, by applying Lemma 2.8 with $\varepsilon = \sqrt{\log N/N}$, we obtain

$$\begin{aligned}
\mathbf{P}\left(||A_{xy}| - \mathbf{E}[|A_{xy}|]| \geq \mathbf{E}[|A_{xy}|]\sqrt{\frac{\log N}{N}}\right) &\leq 2\exp\left(-\frac{\log N}{3N}\mathbf{E}[|A_{xy}|]\right) \qquad (2.5) \\
&= 2\exp\left(-\frac{(N-2)p(1-p)\log N}{3N}\right)
\end{aligned}$$

which tends to 0 as $N \to +\infty$. Therefore, when $N \to +\infty$,

$$|A_{xy}| = \mathbf{E}[|A_{xy}|]\left(1 + O\left(\sqrt{\frac{\log N}{N}}\right)\right) = (1+o(1))Np(1-p). \qquad (2.6)$$

Hence, defining $z \triangleq 1 - p(1-p)$, we can rewrite (2.4) as

$$p_w \leq 2\left(1 - (1+o(1))p(1-p)\right)^w = 2z^{w(1+o(1))},$$

which is at most $1/N^2$ for $w = (2+\varepsilon)\log(\sqrt{2}N)/(\log 1/z)$ (where $\varepsilon > 0$ can be taken arbitrarily small). Applying Lemma 2.9, we conclude that the theorem statement holds.
□

2.3.2 Bound for $\mathcal{G}(N,p)$ with $p = o(1)$

When $p = o(1)$, we cannot derive a bound for the DMD as we did in the proof of Theorem 2.10 because the distinguishing sets A_{xy} and A_{yx} defined by (2.3) are not large enough: more specifically, when we plug (2.6) in (2.4), if $p = o(1)$, we obtain a trivial bound on the probability p_w that a random set of cardinality w is a DRS. Therefore, for the case $p = o(1)$, we have to define different distinguishing sets.

Case 1: $p = \Theta(N^{-i/(i+1)})$ for $i \in \mathbb{N}^+$

We start with the case in which $p = \Theta(N^{-i/(i+1)})$ for some $i \in \mathbb{N}^+$. Note that, with this notation, $i = 0$ yields the case $p = \Theta(1)$ that has been studied in Section 2.3.1. The regimes of p between $\Theta(N^{-i/(i+1)})$ and $\Theta(N^{-(i-1)/i})$ for all $i \geq 1$ will be studied in Case 2 below.

Theorem 2.11. *Suppose that* $p = \Theta(N^{-i/(i+1)})$ *for some* $i \in \mathbb{N}^+$. *Let*

$$c = c(N) \triangleq \frac{\delta^{i+1}}{N} \tag{2.7}$$

and let $z \triangleq 1 - e^{-c}(1 - e^{-c}) = (e^{-c})^2 + (1 - e^{-c})$. *Let* $\mathcal{G} \in \mathcal{G}(n, p)$. *Then a.a.s.*

$$\mathrm{DMD}(\mathcal{G}) \leq \frac{2\log(\sqrt{2}N)}{\log(1/z)}. \tag{2.8}$$

Remark 2.12. *If* $p = \Theta(N^{-i/(i+1)})$ *for some* $i \in \mathbb{N}^+$, *then* $c \triangleq \delta^{i+1}/N = \Theta(1)$ *and therefore* $z = \Theta(1)$. *Indeed,*

$$p = \Theta(N^{-i/(i+1)}) \Rightarrow p^{i+1} = \Theta(N^{-i}) \Rightarrow \frac{p^{i+1}N^{i+1}}{N} = \Theta(1) \Rightarrow c = \frac{\delta^{i+1}}{N} = \Theta(1).$$

Consequently, the order of bound (2.8) *is* $\log(N)$.

We will use the following lemma which formalizes an expansion property of random networks.

Lemma 2.13 (Expansion property of $\mathcal{G}(N, p)$). *Let* $\zeta = \zeta(N)$ *be a function tending to infinity with* N *such that* $p \geq \zeta \log(N)/N$. *Let* $i \in \mathbb{N}^+$ *such that* $\delta^i = o(N)$. *Then the following properties hold a.a.s. for* $\mathcal{G}(N, p)$:

(i) for every $x \in V$

$$|S(x, i)| = \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i \tag{2.9}$$

and for every $V' \subseteq V$ *with* $|V'| = 2$

$$|S(V', i)| = 2\left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i; \tag{2.10}$$

(ii) for every $x, y \in V$ $(x \neq y)$,

$$|S(x,i) \backslash S(y,i)| = \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i. \tag{2.11}$$

We postpone the proof of this lemma to Section 2.3.4. In the proof of Theorems 2.11 and 2.21 we will use 2.9 and 2.11. 2.10 is actually only needed to prove 2.11 (see Section 2.3.4).

Corollary 2.14. *Let* $i \in \mathbb{N}^+$ *such that* $\delta^i = o(N)$*, then* $|N(x,i)| = o(N)$.

Proof. By Lemma 2.13 and the definition of $N(x,i)$ (see Definition 2.7), we have that

$$\begin{aligned} |N(x,i)| &= 1 + \sum_{j=1}^{i} |S(x,j)| = 1 + \sum_{j=1}^{i} \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^j}{N}\right)\right)\delta^j \\ &= \delta^i + o(\delta^i) = o(N). \end{aligned}$$

□

Proof. [Theorem 2.11] The structure of the proof follows that of Theorem 2.10: the only difference is in definition of the distinguishing sets A_{xy} and A_{yx} and in the estimation of their cardinalities.

For $u \neq v$, let A_{uv} be the set of nodes that are at distance $(i+1)$ from u and at distance at least $(i+2)$ from v, i.e.,

$$A_{uv} \triangleq S(u, i+1) \backslash N(v, i+1) = \{z \in V : d(z,u) = (i+1), d(z,v) > (i+1)\}. \tag{2.12}$$

Let $x \neq y$. A_{xy} and A_{yx} are distinguishing sets for x and y. In fact, if $u \in A_{xy}$ and $v \in A_{yx}$, we have

$$d(x,u) - d(x,v) \leq (i+1) - (i+2) = -1$$

and

$$d(y,u) - d(y,v) \geq (i+2) - (i+1) = 1,$$

hence x and y are distinguished by u and v.

We now estimate $|A_{xy}|$ using Lemma 2.13(i). Note that Lemma 2.13 requires the existence of a function $\zeta(N)$ tending to infinity with N and such that $p \geq \zeta \log(N)/N$ and ζ tends to infinity with N. Such a function exists because $p = \Theta(N^{-i/(i+1)})$ implies $p = \omega(\log(N)/N)$. For example, we could take $\zeta = N^{1/(i+2)}/\log N$. Note also that, by Remark 2.12, $\delta^i = o(N)$, hence all the hypotheses of Lemma 2.13 are satisfied.

We have

$$\mathbf{E}[|A_{xy}|] = \sum_{v \in V} \mathbf{P}(d(v,x) = (i+1), d(v,y) \geq (i+2)) \tag{2.13}$$

If $v \in N(x,i)$, $\mathbf{P}(d(v,x) = (i+1)) = 0$ and if $v \in N(y,i)$, $\mathbf{P}(v \notin N(y,i+1)) = 0$. Therefore, in (2.13), all terms with $v \in N(x,i) \cup N(y,i)$ are equal to 0 and (2.13) becomes

$$\mathbf{E}[|A_{xy}|] = \sum_{v \in V \setminus (N(x,i) \cup N(y,i))} \mathbf{P}(d(v,x) = (i+1), d(v,y) \geq (i+2)) \tag{2.14}$$

We observe that, for $v \notin N(x,i) \cup N(y,i)$, $d(v,x) = (i+1)$ if and only if there is an edge between v and a node in $S(x,i)$, whereas $d(v,y) \geq (i+2)$ if and only if there is no edge between v and any node in $S(y,i)$. Hence the two events happen jointly if and only if there is an edge between v and a node in $S(x,i) \setminus S(y,i)$ (which we call the event $\mathcal{E}_1$) and there is no edge between v and any node in $S(y,i)$ (which we call the event $\mathcal{E}_2$). $\mathcal{E}_1$ and $\mathcal{E}_2$ are independent because they depend on disjoint sets of possible edges. Furthermore, for $v \notin N(x,i) \cup N(y,i)$, $\mathbf{P}(\mathcal{E}_1) = (1 - (1-p)^{|S(x,i) \setminus S(y,i)|})$ and $\mathbf{P}(\mathcal{E}_2) = (1-p)^{|S(y,i)|}$.

Therefore, following the derivation of Bollobás et al. [2013], we can rewrite (2.14) as

$$\begin{aligned}
\mathbf{E}[|A_{xy}|] &= \sum_{v \in V \setminus (N(x,i) \cup N(y,i))} (1 - (1-p)^{|S(x,i) \setminus S(y,i)|})(1-p)^{|S(y,i)|} \\
&\overset{(a)}{=} N(1+o(1))(1 - (1-p)^{|S(x,i) \setminus S(y,i)|})(1-p)^{|S(y,i)|} \\
&\overset{(b)}{=} N(1+o(1))(1 - (1-p)^{(1+O(\frac{1}{\sqrt{\zeta}})+O(\frac{\delta^i}{N}))\delta^i})(1-p)^{(1+O(\frac{1}{\sqrt{\zeta}})+O(\frac{\delta^i}{N}))\delta^i} \\
&\overset{(c)}{=} (1+o(1))e^{-c}(1-e^{-c})N, \qquad (2.15)
\end{aligned}$$

where (a) holds by Corollary 2.14; (b) follows from (2.9) and (2.11) in Lemma 2.13; in (c) we used the approximation $(1-p) = \exp(-p + O(p^2))$ for $p \to 0$ and the fact that $p\delta^i \sim \delta^{i+1}/N = c$.

Using the same technique of the proof of Theorem 2.10, we can use (2.15) to prove the theorem statement. □

Case 2: intermediate regimes

We now want to extend our results to the case in which $p = \Omega(1/N^{1-x})$ for some $x > 0$ but $p \neq \Theta(N^{-i/(i+1)})$ for all $i \in \mathbb{N}$. To do so, we first introduce the parameter $i^\star$.

Definition 2.15. *Let $p = \Omega(1/N^{1-x})$ for some $x > 0$. We denote by $i^\star$ the largest integer such that $\delta^{i^\star} = ((N-1)p)^{i^\star} = o(N)$.*

Remark 2.16. *If $p = \Omega(1/N^{1-x})$ for some $x > 0$, $i^\star$ is well defined. As $\delta^0 = 1 =$*

$o(N)$, there is at least one integer such that $\delta^i = o(N)$ ($i = 0$). Let $x > 0$ such that $p = \Omega(1/N^{1-x})$. We have $\delta = \Omega(N^x)$. Let k be the smallest integer such that $kx > 1$, which exists because $x > 0$, then $\delta^k \neq o(N)$ and $i^\star \leq k - 1$.

Remark 2.17. *In the regime of Case 1, i.e., if $p = \Theta(N^{-i/(i+1)})$ for some $i \in \mathbb{N}$, then $i^\star = i$. In fact, we have that*

$$\delta^i \sim (N^{-i/(i+1)})^i N^i = N^{(-i^2+i(i+1))/(i+1)} = N^{i/(i+1)} = o(N)$$

and

$$\delta^{i+1} \sim (N^{-i/(i+1)})^{(i+1)} N^{i+1} = N^{-i} N^{i+1} = N \neq o(N).$$

We can now define the function c of (2.7) for all regimes of p.

Definition 2.18. *Let $p = \Omega(1/N^{1-x})$ for some $x > 0$ and $i^\star$ as in Definition 2.15. We define*

$$c = c(N) \triangleq \delta^{i^\star+1}/N.$$

The following lemma characterizes the values of $i^\star$ and proves that, when $p \neq \Theta(N^{-i/(i+1)})$, $c \to +\infty$. An illustration of the values of $i^\star$ and of c as functions of p is given in Figure 2.6.

Lemma 2.19 (Properties of $i^\star$ and c)**.** *Let $\delta = (N-1)p \sim pN$ and let $i^\star \in \mathbb{N}$ denote the largest integer such that $\delta^{i^\star} = o(N)$.*

(i) Let $i \in \mathbb{N}^+$. $i^\star = i$ if and only if $p \in o(N^{-(i-1)/i})$ and $p \in \Omega(N^{-i/(i+1)})$.

(ii) If $p \neq \Theta(N^{-i/(i+1)})$ for all $i \in \mathbb{N}$, then $c \to +\infty$.

Proof. (i) We have

$$i^\star \geq i \Leftrightarrow \delta^i \in o(N) \Leftrightarrow p \in o(N^{(1-i)/i}) \Leftrightarrow p \in o(N^{-(i-1)/i})$$

and

$$i^\star \leq i \Leftrightarrow \delta^{i+1} \notin o(N) \Leftrightarrow \delta^{i+1} \in \Omega(N) \Leftrightarrow p \in \Omega(N^{-i/(i+1)}),$$

hence the statement follows.

(ii) By Definitions 2.15 and 2.18, $c = \Omega(1)$. Assume by contradiction that $c = \Theta(1)$. We have

$$c = \Theta(1) \Rightarrow \delta^{i^\star+1} = \Theta(N) \Rightarrow p = \Theta(N^{-i^\star/(i^\star+1)}), \tag{2.16}$$

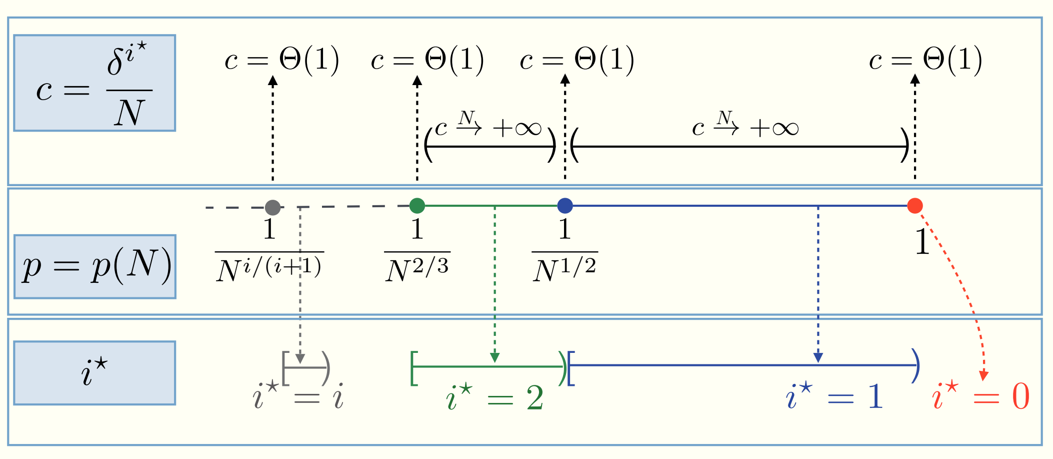


Figure 2.6 – Values of $i^\star$ (see Definition 2.15) and c (see Definition 2.18) as functions of p. For $i \in \mathbb{N}^+$, $i^\star = i$ if and only if $p \in o(N^{-(i-1)/i})$ and $p \in \Omega(N^{-i/(i+1)})$. Furthermore, $c = \Theta(1)$ if and only if $p = \Theta(N^{-i/(i+1)})$ for some $i \in \mathbb{N}$, for all regimes p between any two of these values, $c \to +\infty$.

which contradicts the fact that $p \neq \Theta(N^{-i/(i+1)})$ for all $i \in \mathbb{N}$. Hence, if $p \neq \Theta(N^{-i/(i+1)})$ for every $i \in \mathbb{N}$, $c \to +\infty$.

□

Remark 2.20. *Combining* (2.16) *and Remark 2.12, we have that* $c = \Theta(1)$ *if and only if* $p = \Theta(N^{-i/(i+1)})$ *for some* $i \in \mathbb{N}$*, whereas* $c \to +\infty$ *otherwise.*

We are now ready to state our main result for the case in which $p \neq \Theta(N^{-i/(i+1)})$ for all $i \in \mathbb{N}$, i.e., for $c \to +\infty$.

Theorem 2.21. *Let* $p = \Omega(1/N^{1-x})$ *for some* $x > 0$ *and that* $p \neq \Theta(N^{-i/(i+1)})$ *for all* $i \in \mathbb{N}$*. Let* $i^\star$ *and* c *be defined as in Definitions 2.15 and 2.18. Then*

$$\mathrm{DMD}(\mathcal{G}) \leq 2\left(\frac{\delta^{i^\star}}{N} + e^{-c}\right)^{-1} \log(2N).$$

Proof. Let $x, y \in V$ with $x \neq y$. As in the proofs of Theorems 2.10 and 2.11 above, we want to find large enough distinguishing sets for x and y. We will use the sets A_{xy}, A_{yx} defined by (2.12) with $i = i^\star$, together with other sets B_{xy}, B_{yx} that we will define in the following.

Let us first explain why the distinguishing sets A_{xy}, A_{yx} defined above are not sufficiently large to derive a bound in the present case. To estimate $|A_{xy}|$ via Lemma 2.8, we need that $\mathbf{E}[|A_{xy}|] \to +\infty$ (see (2.5) and (2.6) in the proof of Theorem 2.10 for an example of application of Lemma 2.8). As $c \to +\infty$, we have that $\mathbf{E}[|A_{xy}|] = (1 +$

$o(1))e^{-c}(1-e^{-c})N \to +\infty$ if and only if $Ne^{-c} \to +\infty$, which in turn happens if and only if $c \leq x\log(N)$ with $x < 1$. If this is not the case, in order to obtain an upper bound on the DMD, we need to consider larger sets of distinguishing nodes. In the regime of p we consider, it is possible that $c \geq \log(N)$.[4] For this reason, we will both consider the sets A_{xy}, A_{yx} defined by (2.12) and the sets B_{xy}, B_{yx} defined as follows.

Let B_{uv} be the set of the nodes that are at distance $i^\star$ from u and at distance at least $i^\star+1$ from v, i.e.,

$$B_{uv} \triangleq S(u,i^\star)\backslash N(v,i^\star) = \{z : d(z,u) = i^\star, d(z,v) > i^\star\}.$$

B_{xy} and B_{yx} are distinguishing sets for x and y: If $u \in B_{xy}$ and $v \in B_{yx}$, we have

$$d(x,u) - d(x,v) = i^\star - d(x,v) < 0$$

$$d(y,u) - d(y,v) = d(y,u) - i^\star > 0,$$

therefore $d(x,u) - d(x,v) \neq d(y,u) - d(y,v)$.

We now estimate $|B_{xy}| = |B_{yx}|$ applying Lemma 2.13 with $\zeta = \omega(\log^2 N)$: as $p = \Omega(1/N^{1-x})$ for some $x > 0$ we have $p = \omega(\log^3(N)/N)$ and $p \geq \zeta\log(N)/N$. Furthermore, by definition of $i^\star$, $\delta^{i^\star} = o(N)$.

Call V_y^+ the set of nodes that are at distance strictly larger than $i^\star$ from y and V_y^- the set of nodes at distance strictly smaller than $i^\star$ from y. If $S(y,i^\star)^{\mathsf{c}}$ denotes the complement set of $S(y,i^\star)$, we have therefore that $S(y,i^\star)^{\mathsf{c}} = V_y^+ \cup V_y^-$ and

$$|S(x,i^\star)\backslash S(y,i^\star)| = |S(x,i^\star) \cap S(y,i^\star)^{\mathsf{c}}| = |S(x,i^\star) \cap V_y^+| + |S(x,i^\star) \cap V_y^-|.$$

Estimating $|S(x,i^\star)\backslash S(y,i^\star)|$ via Lemma 2.13(ii), we obtain

$$|B_{xy}| = |S(x,i^\star) \cap V_y^+| = |S(x,i^\star)\backslash S(y,i^\star)| - |S(x,i^\star) \cap V_y^-| = \delta^{i^\star}(1+o(1)),$$

where we used that $(S(x,i^\star) \cap V_y^-) \subseteq V_y^-$ and that, by Lemma 2.13(i), $|V_y^-| = 1 + \delta + \delta^2 ... + \delta^{i^\star-1} + o(\delta^{i^\star-1}) = o(\delta^{i^\star})$.

Note that $|B_{xy}| = \Omega(\sqrt{N})$ because

$$\log_N \delta^{i^\star} = \frac{\log \delta^{i^\star}}{\log N} = \frac{i^\star \log \delta}{\log(\delta^{i^\star+1}/c)} = \frac{i^\star \log \delta}{(i^\star+1)\log\delta - \log c} \overset{(a)}{\geq} \frac{i^\star}{i^\star+1} \geq \frac{1}{2},$$

where (a) holds because, by Lemma 2.19, $c =\to +\infty$.

---

[4] Take for example $p = N^{-i/(i+1)}\log(N)^{1/(i+1)}$ for $i \geq 1$. In this case, it is easy to verify that $i^\star = i$ and $c = \delta^{i+1}/N = \log(N)$.

Remember that, to derive our bound, we want to both use the sets A_{xy}, A_{yx} defined by (2.12) with $i = i^\star$, and the sets B_{xy}, B_{yx}. We have that $A_{xy} \cap B_{xy} = \emptyset$, hence $|A_{xy} \cup B_{xy}| = |A_{xy}| + |B_{xy}|$. Following the argument of Bollobás et al. [2013], we now estimate $|A_{xy}| + |B_{xy}|$.

As explained at the beginning of this proof, if $c \le x\log(N)$ with $x < 1$, we can estimate $|A_{xy}|$ via Lemma 2.8 as in the proof of Theorem 2.11 and we have $|A_{xy}| = (1+o(1))e^{-c}(1-e^{-c})N$. If, instead, $c \to +\infty$ faster than $x\log(N)$ for all $x > 1$ we have, in particular, that $c > 0.51\log(N)$ and

$$\mathbf{E}[|A_{xy}|] \le e^{-c}N \le e^{-0.51\log(N)}N < N^{0.49}$$

and the contribution of $|A_{xy}|$ is negligible with respect to $|B_{xy}| = \Omega(\sqrt{N})$. Hence, for both regimes of c, when $N \to +\infty$,

$$|A_{xy}| + |B_{xy}| = (1+o(1))(e^{-c}(1-e^{-c})N + \delta^{i^\star}).$$

Let $W \subseteq V$ be a set sampled uniformly at random from the collection of all subsets of V of cardinality w. We now want to upper bound the probability p_w that W does not distinguish two nodes x and y. Since both pairs of sets A_{xy}, A_{yx} and of sets B_{xy}, B_{yx} are distinguishing sets for x and y,

$$\begin{aligned}
p_w &\le \mathbf{P}\Big(\{\nexists (v,w) \in (A_{xy} \cap W) \times (A_{yx} \cap W)\} \cap \\
&\qquad\qquad \{\nexists (v,w) \in (B_{xy} \cap W) \times (B_{yx} \cap W)\}\Big) \\
&= \mathbf{P}(\{W \cap (A_{xy} \cup B_{xy}) = \emptyset\} \cup \{W \cap (A_{xy} \cup B_{yx}) = \emptyset\} \cup \\
&\qquad\qquad \{W \cap (A_{yx} \cup B_{xy}) = \emptyset\} \cup \{W \cap (A_{yx} \cup B_{yx}) = \emptyset\}) \\
&\overset{(a)}{\le} 4\mathbf{P}(\{(A_{xy} \cup B_{xy}) \cap W = \emptyset\}) \overset{(b)}{\le} 4\left(1 - \frac{|A_{xy} \cup B_{xy}|}{N}\right)^w \\
&= 4\left(1 - \frac{|A_{xy}| + |B_{xy}|}{N}\right)^w \\
&= 4\left(1 - (1+o(1))(e^{-c}(1-e^{-c}) + \delta^{i^\star}/N)\right)^w \\
&\le 4\exp\left(-(1+o(1))w(e^{-c}(1-e^{-c}) + \delta^{i^\star}/N)\right),
\end{aligned}$$

where in (a) we used that $|A_{xy} \cap B_{xy}| = |A_{yx} \cap B_{xy}| = |A_{xy} \cap B_{yx}| = |A_{yx} \cap B_{yx}| = \emptyset$, hence $|A_{xy} \cup B_{xy}| = |A_{yx} \cup B_{xy}| = |A_{xy} \cup B_{yx}| = |A_{yx} \cup B_{yx}| = |A_{xy}| + |B_{xy}|$; (b) can be derived as (2.4) in the proof of Theorem 2.10. To conclude, we observe that

$4\exp(-w(e^{-c}(1-e^{-c})+\delta^{i^\star}/N)$ is at most $1/N^2$ for

$$w = (2+\varepsilon)\log(2N)\left(e^{-c}(1-e^{-c})+\delta^{i^\star}/N\right)^{-1},$$

where ε can be taken arbitrarily small. Hence by Lemma 2.9 the theorem statement holds. ☐

2.3.3 Overall Bound on the DMD for $\mathcal{G}(N,p)$

The following theorem sums up the bounds we proved in Theorems 2.10, 2.11 and 2.21.

Theorem 2.22 (Upper bounds for DMD)**.** *Let $\mathcal{G} \in \mathcal{G}(N,p)$. Let $p = \Omega(1/N^{1-x})$ for some $x > 0$ and let $i^\star$ be the largest integer such that $\delta^{i^\star} = o(N)$. Define $c = c(N) \triangleq \delta^{i^\star+1}/N$ and*

$$z \triangleq \begin{cases} 1-p(1-p) = p^2+(1-p) & \text{if } p = \Theta(1), \\ 1-e^{-c}(1-e^{-c}) = (e^{-c})^2+(1-e^{-c}) & \text{if } p = o(1). \end{cases}$$

(i) If $c = \Theta(1)$

$$\mathrm{DMD}(\mathcal{G}) \le \frac{2\log(\sqrt{2}N)}{\log(1/z)}.$$

(ii) If $c \to +\infty$

$$\mathrm{DMD}(\mathcal{G}) \le 2\left(e^{-c}+\frac{\delta^{i^\star}}{N}\right)^{-1}\log(2N).$$

For comparison, we state here the upper bound obtained for the MD by Bollobás et al. [2013].

Theorem 2.23 (Upper bounds for MD. Theorem 3.1 in [Bollobás et al., 2013])**.** *Let $\mathcal{G} \in \mathcal{G}(N,p)$. Let $p = \Omega(1/N^{1-x})$ for some $x > 0$ and let $i^\star$ be the largest integer such that $\delta^{i^\star} = o(N)$. Define $c = c(N) \triangleq \delta^{i^\star+1}/N$ and*

$$q \triangleq \begin{cases} 1-2p(1-p) = p^2+(1-p)^2 & \text{if } p = \Theta(1), \\ 1-2e^{-c}(1-e^{-c}) = (e^{-c})^2+(1-e^{-c})^2 & \text{if } p = o(1). \end{cases}$$

(i) If $c = \Theta(1)$

$$\mathrm{MD}(\mathcal{G}) \le \frac{2\log(N)}{\log(1/q)}.$$

(ii) *If* $c \to +\infty$

$$\mathrm{MD}(\mathcal{G}) \leq \left(e^{-c} + \frac{\delta^{i^\star}}{N}\right)^{-1} \log(N).$$

As we could expect, since $\mathrm{DMD}(\mathcal{G}) \geq \mathrm{MD}(\mathcal{G})$, the bounds of Theorem 2.22 are larger than those of Theorem 2.23. Note, in particular, that

(a) z and q are different and $z \leq q$. When $p = \Theta(1)$, both z and q are, as functions of p, parabolas with vertex in $p = 1/2$, furthermore they equal 1 in both $p = 0$ and $p = 1$. Hence $p = 1/2$ yields the maximum distance between z and q. The case $p = o(1)$ is analogous except that z and q are functions of e^{-c}.;

(b) in the bound *(i)* (respectively, *(ii)*), for DMD there is an additional multiplicative factor of $\sqrt{2}$ (respectively, 2) inside the log;

(c) in the bound *(ii)* for DMD we have an additional multiplicative factor of 2.

Finally, Bollobás et al. [2013] also proved that the MD of $\mathcal{G}(N, p)$ can be lower bounded with bounds that are asymptotic to the upper bounds *(i)* and *(ii)* of Theorem 2.23, from which we can directly deduce lower bounds of the same shape for the DMD.

2.3.4 Expansion Properties of Random Networks

We now prove Lemma 2.13 which we used to prove Theorems 2.11 and 2.21.

Lemma 2.13 (Expansion property of $\mathcal{G}(N, p)$)**.** *Let* $\zeta = \zeta(N)$ *be a function tending to infinity with* N *such that* $p \geq \zeta \log(N)/N$. *Let* $i \in \mathbb{N}^+$ *such that* $\delta^i = o(N)$. *Then the following properties hold a.a.s. for* $\mathcal{G}(N, p)$*:*

(i) *for every* $x \in V$

$$|S(x,i)| = \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i \tag{2.9}$$

and for every $V' \subseteq V$ *with* $|V'| = 2$

$$|S(V',i)| = 2\left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i; \tag{2.10}$$

(ii) *for every* $x, y \in V$ $(x \neq y)$,

$$|S(x,i) \backslash S(y,i)| = \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i. \tag{2.11}$$

Remark 2.24. *These expansion properties are proven in Lemma 2.1 of [Bollobás et al., 2013] where they are stated, together with other properties that are not of direct interest for the results in this chapter, under the additional hypothesis that* $\zeta \le (\log N)^4(\log\log N)^2$*. Our proof, which does not use the latter hypothesis, mostly follows the structure of that of Bollobás et al. [2013]. The main difference is in the estimation of* $|N(V,i)|$ *which we obtain here by induction (see* (2.22) *below).*

Proof. (2.11) follows from (2.9) and (2.10), and on the fact that, for any two sets A and B, $|A \cup B| = |A \backslash B| + |B|$. In particular, we take $A = S(x,i)$ and $B = S(y,i)$.

We now prove that for all $V' \subseteq V$ with $|V'| \le 2$

$$|S(V',i)| = \left(1 + O\left(\frac{1}{\sqrt{\zeta}}\right) + O\left(\frac{\delta^i}{N}\right)\right)\delta^i|V'|,$$

which implies both (2.9) and (2.10).

We will take the following steps:

I) estimate the cardinality of $N(Z,1)$ for $Z \subseteq V$ of arbitrary cardinality;

II) derive an estimation of $N(V',1)$ for $V' \subseteq V$ with $|V'| \le 2$;

III) use I) and II) to recursively derive an estimation for $N(V',i)$ with $i \in \mathbb{N}$ and $|V'| \le 2$;

IV) use III) to derive the desired estimation for the cardinality $S(V',i)$ with $i \in \mathbb{N}$ and $|V'| \le 2$.

We start with step I). Let $Z \subseteq V$ of arbitrary cardinality $z = |Z| \ge 1$, and consider the random variable $X = X(Z) = |N(Z,1)|$. We estimate X using Lemma 2.8. First, we compute the expectation of X:

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{v \in V} \mathbf{P}(v \in N(Z,1)) = N - \sum_{v \in V} \mathbf{P}(v \notin N(Z,1)) \\
&= N - \sum_{v \in V \backslash Z} \mathbf{P}(v \notin N(Z,1)) = N - (N-z)\mathbf{P}(v \notin N(Z,1)) \\
&\overset{(a)}{=} N - (N-z)\left(1 - \frac{\delta}{N-1}\right)^z = N - (N-z)\exp\left(z\log\left(1 - \frac{\delta}{N-1}\right)\right) \\
&\overset{(b)}{=} N - (N-z)\exp\left(-\frac{\delta z}{N-1} + O\left(\frac{(\delta z)^2}{N^2}\right)\right) \\
&= N - (N-z)\exp\left(-\frac{\delta z}{N}\left(1 + O\left(\frac{\delta z}{N}\right)\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&\overset{(c)}{=} N-(N-z)\left(1-\frac{\delta z}{N}\left(1+O\left(\frac{\delta z}{N}\right)\right)+O\left(\left[\frac{\delta z}{N}\left(1+O\left(\frac{\delta z}{N}\right)\right)\right]^2\right)\right)\\
&= N-(N-z)\left(1-\frac{\delta z}{N}+O\left(\frac{(\delta z)^2}{N^2}\right)\right)\\
&= N-\left(N-z-\delta z+\frac{\delta z^2}{N}+O\left(\frac{(\delta z)^2}{N}\right)\right)=z(\delta+1)+O\left(\frac{\delta^2 z^2}{N}\right)\\
&= \delta z\left(1+O\left(\frac{\delta z}{N}\right)\right), \qquad (2.17)
\end{aligned}
$$

where (a) follows from the relation $p=\delta/(N-1)$, (b) holds because, when $x\to 0$, $\log(1+x)=x+O(x^2)$, whereas (c) holds because, for $x\to 0$, $\exp(x)=1+x+O(x^2)$. It follows from (2.17) and Lemma 2.8 (with $\varepsilon=4(|Z|\zeta)^{-1/2}$) that

$$
\begin{aligned}
|N(Z,1)| &= \delta|Z|\Big(1+O(\delta|Z|/N)\Big)+O(\varepsilon\mathbf{E}[|N(Z,1)|])\\
&= \delta|Z|\Big(1+O(\delta|Z|/N)\Big)+O(\zeta^{-1/2}\delta|Z|^{1/2}))\\
&= \delta|Z|\Big(1+O(\delta|Z|/N)+O((\zeta|Z|)^{-1/2})\Big) \qquad (2.18)
\end{aligned}
$$

with probability at least

$$
\begin{aligned}
1-2\exp(-\varepsilon^2|Z|\delta/3) &= 1-2\exp(-16\delta/3\zeta) \overset{\delta\geq\zeta\log(N)}{\geq} 1-2\exp(-16\log(N)/3)\\
&= 1-\frac{2}{N^{16/3}}\geq 1-\frac{1}{N^3}. \qquad (2.19)
\end{aligned}
$$

Next, we move to step II). By applying Lemma 2.8 with $\varepsilon=2\sqrt{\zeta^{-1}}$, the expected number of sets V' with $|V'|\leq 2$ such that

$$
\Big||N(V',1)|-\mathbf{E}[|N(V',1)|]\Big|>\varepsilon\delta|V'|
$$

is

$$
\begin{aligned}
&\sum_{|V'|\leq 2}\mathbf{P}\left(\Big||N(V',1)|-\mathbf{E}[|N(V',1)|]\Big|>\varepsilon\delta|V'|\right)\\
&\leq \sum_{z=1,2}N^z 2\exp\left(\frac{-\varepsilon^2 z\delta}{3+o(1)}\right)\overset{(a)}{\leq}\sum_{z=1,2}N^z 2\exp\left(\frac{-\varepsilon^2 z\zeta\log(N)}{3+o(1)}\right)\overset{(b)}{=}o(1)
\end{aligned}
$$

where (a) holds because, by hypothesis, $\delta\geq\zeta\log(N)$ and (b) because

$$
N^z\exp\left(\frac{-\varepsilon^2 z\zeta\log(N)}{3+o(1)}\right)\sim\frac{N^z}{N^{\varepsilon^2 z\zeta/3}}=\frac{N^z}{N^{4z/3}}=\frac{1}{N^{z/3}}\to 0
$$

for $N\to+\infty$.

Hence, for $|V'| \leq 2$, it holds a.a.s. that

$$\begin{aligned}
|N(V',1)| &= \mathbf{E}[X] + O(\varepsilon \mathbf{E}[X]) \\
&= \delta|V'|\left(1 + O\left(\frac{\delta|V'|}{N}\right)\right) + O\left(\frac{2\delta|V'|}{\sqrt{\zeta}}\right) \\
&= \delta|V'|\left(1 + O\left(\frac{\delta}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) \qquad (2.20)
\end{aligned}$$

and step II) is complete.

For step III) we observe that $\delta^i \leq N$ implies $i \leq \log(N)/\log(\delta)$ and, since $\delta \geq \log(N)$, we have $i \leq \log(N)/\log(\log(N))$. Hence the sets $N(V',i)$ with $|V'| \leq 2$ and $\delta^i \leq N$ are at most $O(N^2 \log(N))$. In view of (2.19), we deduce that the expected number of sets $Z = N(V',i)$ for which (2.18) does not hold is upper bounded by $N^2 \log(N)/N^3 \to 0$, hence (2.18) holds a.a.s.

We are ready to obtain an estimation of $|N(V',i)|$ by induction. Assuming that

$$|N(V',i-1)| = \delta^i|V'|\left(1 + O\left(\frac{\delta^{i-1}}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right), \qquad (2.21)$$

we can use $N(V',i) = N(N(V',i-1),1)$ to obtain that

$$\begin{aligned}
|N(V',i)| &= |N(N(V',i-1),1)| \\
&\overset{(2.20)}{=} \delta|N(V',i-1)|\left[1 + O\left(\frac{\delta|N(V',i-1)|}{N}\right) + O\left(\frac{1}{\sqrt{\zeta|N(V',i-1)|}}\right)\right] \\
&\overset{(2.21)}{=} \delta^i|V'|\left(1 + O\left(\frac{\delta^{i-1}}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right)\left[1 + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta\delta^{i-1}}}\right)\right] \\
&= \delta^i|V'|\left(1 + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right). \qquad (2.22)
\end{aligned}$$

Finally, for step IV), by definition of $S(V',i)$ and $N(V',i)$ we obtain

$$\begin{aligned}
|S(V',i)| &= |N(V',i)| - |N(V',i-1)| \\
&\overset{(2.22)}{=} \delta^i|V'|\left(1 + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) - \delta^{i-1}|V'|\left(1 + O\left(\frac{\delta^{i-1}}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) \\
&= \delta^i|V'|\left(1 + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) - \delta^{i-1}|V'| \\
&= \delta^i|V'|\left(1 + \frac{1}{\delta} + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) \\
&\overset{(a)}{=} \delta^i|V'|\left(1 + O\left(\frac{\delta^i}{N}\right) + O\left(\frac{1}{\sqrt{\zeta}}\right)\right) \qquad (2.23)
\end{aligned}$$

where (a) holds because $\delta \geq \zeta \log(N) \geq \sqrt{\zeta}$. $\square$

2.4 Experimental Results

In this section, we experimentally show that the DMD approximation obtained through the greedy algorithm by Chen et al. [2014] follows, as a function of p, the behavior predicted by the bounds that we derived in Section 2.3. Furthermore, we look at the error probability $\mathcal{P}_e$ of source localization when only a fixed number of sensors is available and we find that the non-monotonicity of the DMD is reflected also in the behavior of $\mathcal{P}_e$.

Finally, we empirically compare the MD and the DMD, first for $\mathcal{G}(N, p)$ networks and then for trees.

2.4.1 The DMD of $\mathcal{G}(N, p)$ Networks

Non-monotonic Behavior of the DMD

We generate several instances of $\mathcal{G}(N, p)$ networks for varying values of p, and we approximate the DMD of the networks with the greedy algorithm proposed by Chen et al. [2014]. We have already seen in Figure 2.5 that the DMD follows closely the ratio $|W_0^{x,y}|/|W_1^{x,y}|$. In Figure 2.5a we also observe that, apart from the main peak, for very small values of p, another local maximum seems to appear. Figure 2.7 depicts a more detailed version of the DMD curve for small values of p. Remember, from Section 2.3.2, that the values of p for which $c = \Theta(1)$, i.e., for which the upper bound we derived reaches its minima, are of the form $p = N^{-i/(i+1)}$. In Figure 2.7, we marked with two vertical lines the points $p = N^{-1/2}$ (rightmost line) and $p = N^{-2/3}$ (leftmost line), where, as expected, DMD reaches a minimum value. Note that for the selected value of N ($N = 2000$), the values $p = N^{-i/(i+1)}$ for $i \geq 3$ do not give connected instances of $\mathcal{G}(N, p)$ networks, hence are not displayed in the figure.

DMD & Error Probability in Source Localization

We show that the DMD of a network indeed reflects the difficulty of source localization. In Section 1.2.3, we introduced several metrics for source localization and, in particular, the success probability $\mathcal{P}_s$ (see (1.3)), i.e., the probability that the estimated source $\widehat{s}$ is equal to the real source $s^\star$. We consider the case in which epidemics spread with deterministic transmission delays and we want to estimate the source looking at the infection times $\{t_s : s \in \mathcal{S}\}$ of a set $\mathcal{S} \subseteq V$ of sensors. As explained in Section 1.2.2, using $\{t_s : s \in \mathcal{S}\}$ we can determine the equivalence class of the source $[s^\star]$. We then produce an estimate $\widehat{s}$ of the source by sampling uniformly at random from $[s^\star]$. In this way, if the source is a singleton, i.e., $[s^\star] = \{s^\star\}$, $\mathcal{P}_s = 1$, otherwise $\mathcal{P}_s = 1/|[s^\star]|$. In

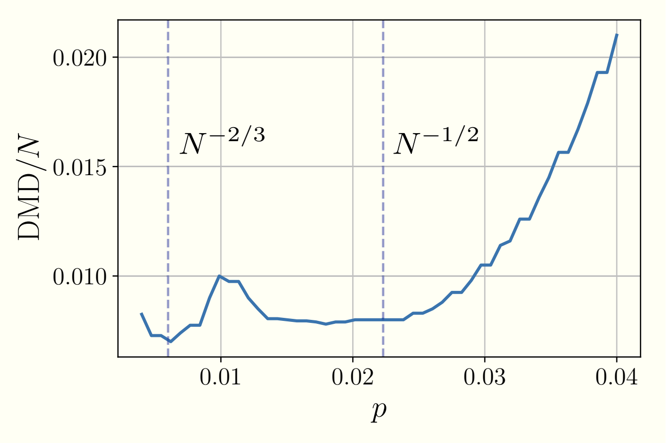


Figure 2.7 – Approximate DMD for small values of p and $N = 2000$. The two vertical lines mark the points $p = N^{-1/2}$ and $p = N^{-2/3}$, where, as predicted by the results of Section 2.3.2, the DMD reaches a minimum value.

order to make the comparison with the DMD easier, we consider here the probability of error $\mathcal{P}_e \triangleq 1 - \mathcal{P}_s$ rather than $\mathcal{P}_s$. The optimization of the sensor choice will be studied in the next chapter and we ignore it here; for the results we display, the sensors were chosen with the LV-Obs algorithm that will be presented in Chapter 4. Figure 2.8a shows the average value (over the possible sources $s^\star$) of $\mathcal{P}_e$ for varying p when the number of sensors is fixed to $|\mathcal{S}| = 0.02 \cdot N$. We see that the curve follows closely that of the DMD (see Figure 2.8c) and, in particular, both local maxima are matched.

Comparison between MD & DMD

In Figure 2.8c, we compare the approximation of DMD obtained with the algorithm of Chen et al. [2014] (DMD) with the approximation of MD obtained with the algorithm of Hauptmann et al. [2012] (MD). Note that the two curves follow closely each other but that the DMD approximation is consistently smaller than the MD approximation. We know that $\text{DMD}(\mathcal{G}) \geq \text{MD}(\mathcal{G})$: This apparently contradictory result is due to the approximation quality of the two algorithms. Note, in particular, that the algorithm of Hauptmann et al. [2012] is a greedy algorithm that starts with the emptyset and iteratively selects the node that minimizes a certain objective function. Also the algorithm of Chen et al. [2014] is greedy but, as at least two nodes are needed to distinguish any two other nodes in a DMD sense, the algorithm loops over all possible initializations $\mathcal{S}_0 = \{v\}$, $v \in V$. For this reason, we consider also an improved (even if more costly in runtime) version of the algorithm of Hauptmann et al. [2012] where we also loop over all possible initializations $\mathcal{S}_0 = \{v\}$, $v \in V$. The resulting (approximated) MD, also depicted in Figure 2.8c, is, as expected, smaller than the one obtained with the algorithm of Hauptmann et al. [2012] and is now very close to the DMD approximation. We conclude that, in the case of $\mathcal{G}(N, p)$ networks the (approximated) values of MD and DMD are very close to each other. From a source localization perspective, this means that when the information on the starting time of the epidemic is not available, the cost-difference in terms of sensors needed to localize the source is very little or even not existent. Of course, this result is highly dependent on the topology (see, e.g., the discussion about trees in Section 2.4.2.)

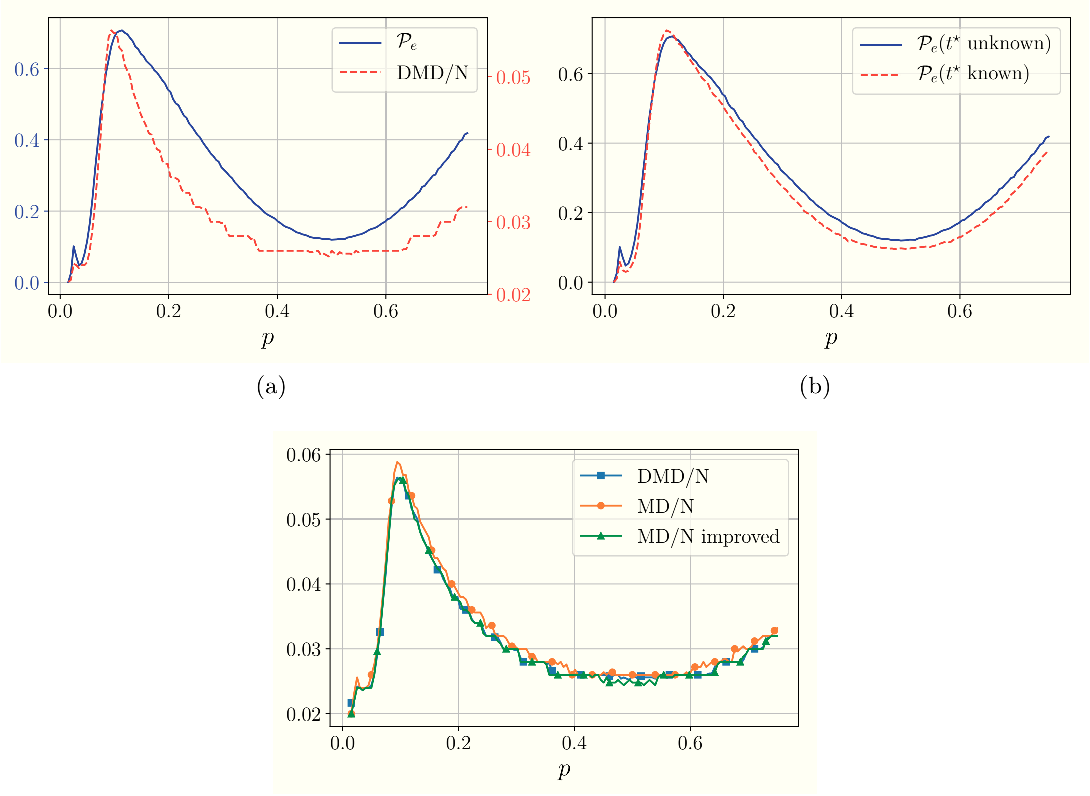
Finally, Figure 2.8b shows again the probability of error $\mathcal{P}_e$ when $t^\star$ is unknown (Figure 2.8a) compared to $\mathcal{P}_e$ when $t^\star$ is known and used to localize the source. In this last case we chose the sensors by stopping each run of the improved algorithm of Hauptmann et al. [2012] described above after the number of $|\mathcal{S}| = 0.02 \cdot N$ sensors was reached. As we could expect, $\mathcal{P}_e$ is in general lower when $t^\star$ is known because more information is available. Nevertheless the improvement with respect to the case in which $t^\star$ is unknown is relatively small, which again shows that when the information on the starting time of the epidemic is not available, the amount of resources needed for source localization does not dramatically increase. Note also that, close to the values of p for which $\mathcal{P}_e$ reaches its maximum, $\mathcal{P}_e$ appears to be smaller when $t^\star$ is known. This is again due to the quality of the algorithm for approximating the MD which seems to perform particularly poorly when larger sets of sensors are needed to detect the source.

2.4.2 The DMD and the MD of Other Random Networks

Trees

For trees, we know that the DMD is equal to the number of leaves (see Proposition 1.6), whereas the MD can be computed with the formula given in Proposition 2.6. We look at how the difference between the MD and the DMD varies with the tree topology. In particular, we consider a family of random trees where, for each node, the number of children N_c is distributed according to a geometric random variable Geom(q) with $0 < q < 1$, i.e., for $i \in \mathbb{N}^+$, $\mathbf{P}(N_c = i) = q(1-q)^{i-1}$. A tree $\mathcal{T}$ is generated starting from one single node and adding, one after the other, some generations of new nodes, i.e., assigning children to the nodes that are currently leaves. For this family of random trees, the DMD is linear in q: This is a consequence of the fact that the DMD equals the number of leaves and that the number of children N_c is geometrically distributed. We give a proof of this fact in the next lines. Let N denote the size of $\mathcal{T}$ after $k \in \mathbb{N}^+$ generations. Let $z \triangleq 1/q$ and denote by $X(i)$ the size of the i^{th} generation in the tree, $X(0) = 1$. We have that $\mathbb{E}[X(i)] = z^i$ and we use the following identity: $z^i - 1 = (z-1)(1 + z + ... + z^{i-1})$. We have

$$N = X(0) + ... + X(k) = 1 + z + z^{k-1} + z^k = z^k + \frac{z^k - 1}{z - 1}$$



(a)

(b)

(c)

Figure 2.8 – $\mathcal{G}(N, p)$ networks with $N = 500$. **(a)**: Probability of error $\mathcal{P}_e$ in source localization as a function of p when the epidemic starting time $t^\star$ is unknown. **(b)**: Probability of error $\mathcal{P}_e$ when $t^\star$ is unknown compared to $\mathcal{P}_e$ when $t^\star$ is known and used to localize the source. **(c)**: Comparison between the approximate DMD obtained with the algorithm of Chen et al. [2014] (DMD), the approximate MD obtained with the algorithm of Hauptmann et al. [2012] (MD) and an improved MD approximation obtained by initializing the algorithm of Hauptmann et al. [2012] in turn with each possible singleton $\{v\} \subset V$ (MD improved).

Hence the number of leaves of the tree $\mathcal{T}$ is

$$z^k = \frac{N + 1/(z-1)}{1 + 1/(z-1)} = \frac{N(z-1)+1}{z} = \frac{N(1-1/q)}{1/q} + O\left(\frac{1}{N}\right)$$

and we can conclude that, as shown in Figure 2.9a,

$$\frac{\mathrm{DMD}(\mathcal{T})}{N} = (1-q) + O\left(\frac{1}{N^2}\right) \sim 1 - q.$$

Figure 2.9a depicts the average values of the MD and the DMD as a function of q.

First, we note that for small q, both the DMD and MD approach N. In fact, when q gets smaller, the tree becomes more similar to a star network for which we know that the DMD is $N-1$ and the MD is $N-2$. Conversely, when q is close to 1 both the MD and DMD approach 0: In this case the tree becomes similar to a path network for which we know that the DMD is 2 and the MD is 1. For intermediate values of q the MD is significantly smaller than the DMD, with the largest difference at around $q = 0.5$. This is an example of a network topology where not knowing the starting time of the epidemic incurs a significantly higher cost in terms of sensors.

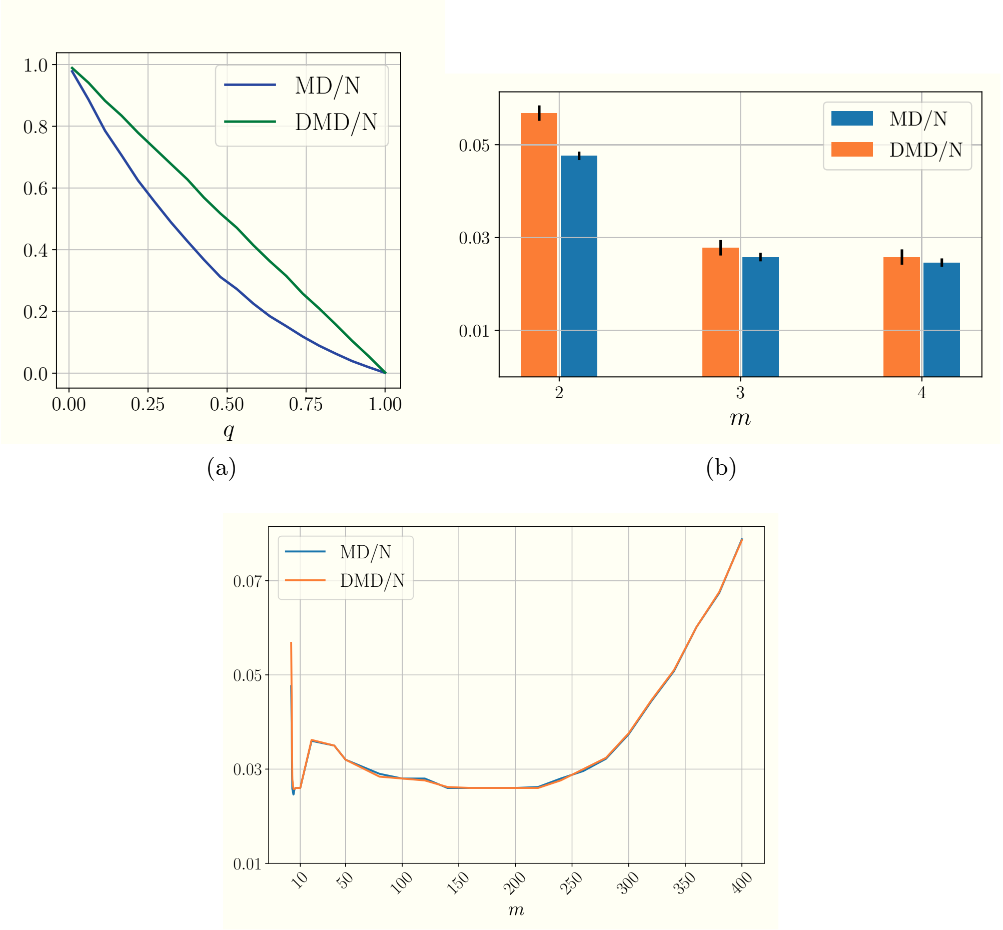
Barábasi-Albert Networks

Figure 2.9c compares the (approximated) MD and the DMD of Barabási-Albert networks with parameter $m \in \{2, 3, 4\}$. Here m is the number of existing nodes in the network to which a new node is connected, hence, for larger m, the network topology gets further from a tree topology. We observe that the average MD is consistently smaller than the average DMD and, as we could expect based on the discussion of the tree-case above, the difference between the two decreases when the parameter m increases.

Finally, Figure 2.9b shows the MD and the DMD for larger values of m. We note that the MD and the DMD curves are very close to each other and, interestingly, they both present a non-monotonic behavior similar to that observed for $\mathcal{G}(N,p)$ random networks. This result shows that the dependency of the MD and the DMD on the density of edges goes beyond the $\mathcal{G}(N,p)$ topology and characterizes a broader class of random networks.

2.5 Discussion

In this chapter, we derived bounds for the DMD for $\mathcal{G}(N,p)$ networks. These bounds are non-monotonic in the parameter p and we experimentally showed that the DMD seems to follow the same non-monotonic behavior. Moreover, we empirically showed that for these networks, the DMD is very close to the MD, which implies that the cost, in terms of the number of sensors needed for source localization, of not having access to the starting time



(a)

(b)

(c)

Figure 2.9 – **(a)**: MD and DMD for random trees of $N = 1000$ nodes. The number of children of each node is distributed as Geom(q). **(b-c)**: MD and DMD for Barabási-Albert networks of $N = 500$ nodes and parameter m.

of the epidemic, is very low.

For future work, several extensions are of interest. First, from a theoretical perspective, it would be valuable to identify other network families for which similar bounds can be found. Natural candidates would be Random Geometric Networks [Penrose, 2003]. This model shares with the $\mathcal{G}(N, p)$ model a well-concentrated degree distribution. Furthermore, the presence of an edge connecting a node y to a node x, is, similarly to the $\mathcal{G}(N, p)$ case, a binomial random variable (depending, however, on the position of node x).

Second, it would be interesting to experimentally study the DMD of more complex networks containing $\mathcal{G}(N, p)$-like clusters, e.g., the stochastic block model [Holland et al., 1983]. In this case, it would be interesting to study the effect on the DMD of the interplay between the p parameter of the $\mathcal{G}(N, p)$ components and the between-clusters connectivity.

Finally, as for $\mathcal{G}(N,p)$ the MD and the DMD seem to follow a very similar behavior as functions of p, a natural extension of this work would investigate the similarity between the minimum-size RS and the minimum size DRS of a network. This would lead to understand if there are nodes, or group of nodes, that emerge as being key to distinguishing among other nodes both in RS's and DRS's.

3 Sensor Placement on Trees

We defined a sensor to be a node for which, when an epidemic spreads in a network, the infection time is known (see Definition 1.1).

If we want to localize the source of an epidemic and we can choose only K nodes as sensors, what is the optimal choice? Answering this question is not easy, even in the simple case of deterministic transmission delays: in Section 1.2.2 we proved that, for general networks, this is actually a NP-hard problem.

In this chapter we focus on the special case of trees where the uniqueness of the path connecting any two nodes makes sensor placement easier to tackle. Moreover we assume the variance of the transmission delays to be *low* in a sense that we will soon specify. Our main results are two distinct polynomial-time algorithms for finding the K sensors that optimize the probability of success in source localization and the expected distance between the real source and the estimated one.

Our approach and our terminology are developed based on the link between sensor placement and the double-resolving-set (DRS) problem described in Section 1.2.2. This provides a tractable and intuitive framework for optimizing sensor placement.

The results presented in this chapter were published in [Celis, Pavetić, Spinelli, and Thiran, 2015].

3.1 Overview

Building on the definition of DRS, we begin by introducing the concept of *node resolvability*, and we show that the performance of a sensor set, with respect to the success probability $\mathcal{P}_s$, is directly linked to the number of *resolved* nodes (see Section 3.2). Our main results are in Sections 3.3 and 3.4. For the case of a tree $\mathcal{T}$ of size N with uniform prior on the identity of the source, we design an $O(NK^2)$ dynamic-programming algorithm to find K sensors that maximize the number of resolved nodes and, as a result, the success probability. Also for the minimization of the expected error distance $\mathcal{D}_e$, we show that if $\mathcal{G}$ is a tree with bounded node degree, an optimal sensor set $\mathcal{U}$ can be found via a polynomial-time algorithm. In Section 3.5, we generalize our results to (i) the case in which sensors have a non-unit cost and the budget limits the maximum total cost of the sensor set and (ii) the case of general priors on the identity of the source.

3.2 Preliminaries

3.2.1 Model

We consider the model described in Section 1.2.1. In this chapter, we look only at the case of deterministic epidemics ($X_{uv} = w_{uv}$ for every $uv \in E$) and non-deterministic epidemics with bounded noise (see Proposition 3.1 in Section 3.2 for the details).

Given a budget $K \in \mathbb{N}$, we study the problem of finding a set $\mathcal{U} \subseteq V$ of size K such that observing the infection times of the nodes in $\mathcal{U}$ optimizes a given metric.

We consider two possible metrics for quantifying the performance of source localization:

1. the *success probability*, i.e., $\mathcal{P}_s \triangleq \mathbf{P}(\widehat{v} = v^\star)$;
2. the *expected distance between the real source* $v^\star$ *and the estimated source* $\widehat{v}$, i.e., $\mathcal{D}_e \triangleq \mathbf{E}[d(v^\star, \widehat{v})]$, where d is the weighted distance between two nodes in the network.

We have already seen that optimizing these two metrics can require different sets of sensors (see Section 1.2.3).

Recall that, in a deterministic epidemic, if a node v gets infected at time t_v, it infects each non-infected neighbor u at time $t_u = t_v + w_{uv}$ where $w_{uv} \in \mathbb{R}^+$ is the weight of edge uv. Then it is clear that, based on the observation vector $\mathbf{t}_{u_1}$, we can identify a set of nodes that are candidate sources (see Definition 1.10 in Chapter 1). Specifically, if the source of the diffusion is $v^\star$, based on the observation vector $\mathbf{t}_{u_1}$, we identify all nodes in $[v^\star]$ as *candidate source nodes* because their distance vectors are equal to $\mathbf{t}_{u_1}$. Then we can produce an estimated source $\widehat{v}$ by by sampling the conditional distribution $\pi|_{[v^\star]}$, where π denotes the prior on the identity of the source. If no prior is available, we sample uniformly from $[v^\star]$.

If the transmission delays are random but do not deviate much from their average values, we are always able to identify the equivalence class to which the real source belongs by finding the class $[v]$ that minimizes the distance between the distance vector $\mathbf{d}_{v,u_1}$ and the observation vector $\mathbf{t}_{u_1}$. In other words, the estimator $\widehat{v}$ that we have just defined tolerates a bounded amount of noise in the transmission delays.

3.2.2 Noise Tolerance

Proposition 3.1 (Noise tolerance)**.** *Let $\mathcal{T}(V,E)$ be a tree, $\mathcal{U} \subseteq V$ a set of sensors and call Δ the maximum distance between a node in $\mathcal{U}$ and a node in V, $\bar{w}$ the maximum edge weight in $\mathcal{T}$. If, for every $uv \in E$, the transmission delay X_{uv} is a random variable such that $X_{uv} \in [(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}]$ and $\varepsilon < \bar{w}/\Delta$, the equivalence class of the source is $[v^\star] = [v]$ such that*

$$[v] = \arg\min_{z \in V} \|\mathbf{d}_{z,u_1} - \mathbf{t}_{u_1}\|_\infty.$$

Proof. First, we prove that for every v, z such that $[v] \neq [z]$, $\|\mathbf{d}_{v,u_1} - \mathbf{d}_{z,u_1}\|_\infty \geq 2\bar{w}$. Since $[v] \neq [z]$ there exists $u \in \mathcal{U}$ such that $d(v,u) - d(v,u_1) \neq d(z,u) - d(z,u_1)$. Take $v', z' \in \mathcal{P}(u,u_1)$ (possibly equal to v, z or to u, u_1) such that $d(v',u) - d(v',u_1) = d(v,u) - d(v,u_1)$ and $d(z',u) - d(z',u_1) = d(z,u) - d(z,u_1)$. More precisely, v' is the last common node in the ordered paths $\mathcal{P}(v,u)$ and $\mathcal{P}(v,u_1)$ (which necessarily lies on $\mathcal{P}(u,u_1)$ because $\mathcal{T}$ is a tree) and, analogously, z' is the last common node in the paths $\mathcal{P}(z,u)$ and $\mathcal{P}(z,u_1)$. Necessarily $v' \neq z'$ (otherwise v and z would not be distinguished by the pair u, u_1). Without loss of generality we can assume $d(v',u) < d(z',u)$, i.e., going from u to u_1 we pass first through v' and then through z'. We have $d(v',u_1) - d(z',u_1) > d(v',z') \geq \bar{w}$ and analogously $d(z',u) - d(v',u) > d(v',z') \geq \bar{w}$. By combining the last two we obtain

$$d(v,u_1) - d(z,u_1) - d(v,u) + d(z,u) = (d(v',u_1) - d(z',u_1)) + (d(z',u) - d(v',u)) \geq 2\bar{w}$$

and we can conclude $\|\mathbf{d}_{v,u_1} - \mathbf{d}_{z,u_1}\|_\infty \geq 2\bar{w}$.

Let $t_{u'}$ be the infection time of $u' \in \mathcal{U}$. When the source is $v^\star$ we have

$$t_{u'} - t^\star \leq d(v^\star, u')(1+\varepsilon). \tag{3.1}$$

Moreover,

$$t_{u'} - t^\star \geq d(v^\star, u')(1-\varepsilon). \tag{3.2}$$

Combining inequalities (3.1) and (3.2) for u' being, respectively, u and u_1 and calling t_1 (resp., t_u) the infection time of the reference sensor u_1 (resp., u), we have

$$|t_u - t_1 - d(v^\star,u) + d(v^\star,u_1)| \leq \varepsilon(d(v^\star,u) + d(v^\star,u_1)) \leq 2\varepsilon\Delta.$$

Since, for every $v \in [v^\star]$, $\mathbf{d}_{v,u_1} = \mathbf{d}_{v^\star,u_1}$, we conclude that for every $v \in [v^\star]$, $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 2\varepsilon\Delta$.

Take now $v \notin [v^\star]$ and assume by contradiction that $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 2\varepsilon\Delta$. Using the triangular inequality and the hypothesis $\varepsilon < \frac{\bar{w}}{\Delta}$ we have

$$\|\mathbf{d}_{v^\star,u_1} - \mathbf{d}_{v,u_1}\|_\infty \leq \|\mathbf{d}_{v^\star,o_1} - \mathbf{t}_{u_1}\|_\infty + \|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 2\varepsilon\Delta < 2\bar{w},$$

which gives a contradiction with $\|\mathbf{d}_{v^\star,u_1} - \mathbf{d}_{v,u_1}\|_\infty > 2\bar{w}$. Hence for every $v \notin [v^\star]$, $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty > \|\mathbf{d}_{v^\star,u_1} - \mathbf{t}_{u_1}\|_\infty$.

□

3.2.3 Resolved Nodes

Building on the definition of distance vector (see Definition 1.8 in Chapter 1), we can introduce the notions of *resolved* node and *unresolved* node.

Definition 3.2 (Resolved & unresolved nodes)**.** *Let $\mathcal{G}$ be a network, $\mathcal{U} \subseteq V$, $|\mathcal{U}| = K \geq 2$ a set of sensors. Fix $u_1 \in \mathcal{U}$. A node z is resolved by a set $\mathcal{U}$ if and only if $\mathbf{d}_{z,u_1} \neq \mathbf{d}_{v,u_1}$ for all $v \in V$, $v \neq u$, and unresolved otherwise.*

In other words, a resolved node is a node z that can be distinguished from all other nodes based on the relative distances between z and the sensors $\mathcal{U}$. As a consequence of Lemma 1.9, Definition 3.2 does not depend on the choice of the reference sensor u_1 and z is resolved by $\mathcal{U}$ if and only if $[z]_\mathcal{U} = \{z\}$.

In the following lemma we make some observations that are key to solve sensor placement on trees. The three statements are illustrated in Figure 3.1.

Lemma 3.3. *Let $\mathcal{T}$ be a tree with N nodes, $\mathcal{U} \subseteq V$, $|\mathcal{U}| \geq 2$.*

(i) Let $z \in \mathcal{U}$ be a non-leaf node or $z \in V \backslash \mathcal{U}$. If there are no $u_1, u_2 \in \mathcal{U}$, $u_1, u_2 \neq z$, such that $z \in \mathcal{P}(u_1, u_2)$, then z is not resolved by $\mathcal{U}$;

(ii) let $z \in V$ a non-leaf node and root $\mathcal{T}$ at z. z is resolved if and only if every subtree $\mathcal{T}_c$ rooted at a child c of z contains at least one node of $\mathcal{U}$;

(iii) a leaf ℓ is resolved if and only if $\ell \in \mathcal{U}$.

Proof.

(i) Suppose first that $z \in V \backslash \mathcal{U}$. If there do not exist $u_1, u_2 \in \mathcal{U}$, such that $z \in \mathcal{P}(u_1, u_2)$ it means that all nodes in $\mathcal{U}$ are in a subtree rooted at a neighbor of z, say c, and not containing z itself. Hence it is easy to see that z is equivalent to c and is not resolved. Next, suppose that $z \in \mathcal{U}$ is a non-leaf node. If there do not exist $u_1, u_2 \in \mathcal{U}$, $u_1, u_2 \neq z$, such that $z \in \mathcal{P}(u_1, u_2)$, it means that there is at least a

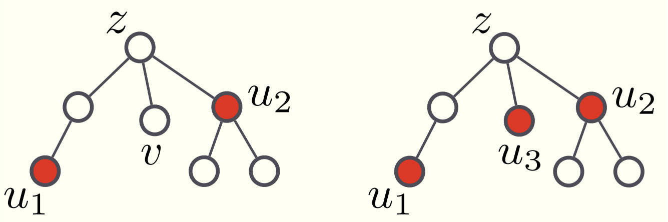


Figure 3.1 – Illustration of Lemma 3.3. Red nodes represent sensors. On the left, illustration for statement *(i)*: v does not lie on the path between the two sensors and is not resolved; u_2 is also not resolved: in fact it is equivalent to its two leaf neighbors. On the right, illustration for statements *(ii)* and *(iii)*: the tree rooted at z contains a sensor in each subtree, hence z is resolved; leaves u_1 and u_3 are resolved, the other leaves are not.

neighbor of z, say c, such that no node of $\mathcal{U}$ is contained in the subtree rooted at c and not containing z. Hence z is not resolved because it is equivalent to c.

(ii) Let z be a resolved non-leaf node. By contradiction, let z have a neighbor c such that the subtree rooted at c and not containing z has empty intersection with $\mathcal{U}$. Then, as in (i), c is equivalent to z, giving a contradiction with the fact that z is resolved. For the backward direction take a non-leaf node z and a node v in a subtree $\mathcal{T}_c$ rooted at a neighbor c of z. Since z is not a leaf, it has at least one other neighbor $h \neq c$. To resolve the pair (z, v) it is then enough to take a sensor in $\mathcal{T}_c$ and one in $\mathcal{T}_h$.

(iii) Let $\ell \in \mathcal{U}$ be a leaf. Since $|\mathcal{U}| > 1$ there exists $t \in \mathcal{U}$, $t \neq \ell$. Remember that $\mathcal{T}$ is a tree: for every $x \in V$, $x \neq \ell, t$, there exists a node y such that $y \in \mathcal{P}(\ell, t) \cap \mathcal{P}(x, \ell) \cap \mathcal{P}(x, t)$, with possibly $y = x$ or $y = t$. Then, $d(\ell, t) - d(\ell, \ell) = d(\ell, t) > d(y, t) - d(y, \ell) = d(x, t) - d(x, \ell)$, implying that ℓ is resolved by $\mathcal{U}$. Suppose next that $\ell \notin \mathcal{U}$. Then ℓ is equivalent to its only neighbor and is not resolved by $\mathcal{U}$.

□

3.3 Success Probability Maximization

We first consider the maximization of the success probability. Maximizing the success probability is equivalent to minimizing the *error probability* $\mathcal{P}_e = \mathbf{P}(\widehat{s} \neq s^\star)$. We express our results in terms of this last metric.

From (1.3) in Chapter 1 we derive

$$\mathcal{P}_e(\mathcal{U}) = \frac{1}{N} \sum_{[u]_{\mathcal{U}} \subseteq V} (|[u]_{\mathcal{U}}| - 1) = 1 - \frac{q}{N} \tag{3.3}$$

where q denotes the number of equivalence classes.

It is clear that $\mathcal{P}_e$ is minimized if the number of equivalence classes is maximized.

We proved in Proposition 1.6 in Chapter 1 that, if the network is a tree $\mathcal{T}$ and all leaves are sensors, $\mathcal{P}_e = 0$. Moreover, we have seen in Proposition 1.12 that when the budget K is smaller than the set of leaves the optimal sensor placement is contained in the leaf set and the minimum value of $\mathcal{P}_e$ is strictly larger than 0. This suggests that, given a tree and a sensor set, if we root the tree at an arbitrary node it is possible to compute $\mathcal{P}_e$ as the sum of the probabilities of error of the different subtrees. Building on this observation we prove that, for a tree $\mathcal{T} = \mathcal{T}(V, E)$ with $|V| = N$ and a budget $K \in \mathbb{N}$, a set $\mathcal{U}_{\text{opt}}$ with $|\mathcal{U}_{\text{opt}}| = K$ that minimizes $\mathcal{P}_e$ can be found with a recursive algorithm of total runtime $O(NK^2)$.

Theorem 3.4. *Let $\mathcal{T}$ be a tree with N nodes and ℓ leaves and let the prior π be uniform. If $K \geq \ell$, the leaves form an optimal sensor set. If $K \in \{2, \dots, \ell - 1\}$, there exists an algorithm that finds $\mathcal{U}_{\text{opt}} \in \text{argmin}_{|\mathcal{U}|=K} \mathcal{P}_e(\mathcal{U})$ in time $O(NK^2)$.*

Correctness. The statement is trivial for $K \geq \ell$ as the set of leaves resolves all the nodes. If $2 \leq K < \ell$, call $\mathcal{T}_r$ the tree obtained rooting $\mathcal{T}$ at an arbitrary non-leaf node r. We claim that $\mathcal{U}_{\text{opt}}^K$ is obtained through the main function of Algorithm 2, i.e., by computing $\textsc{OptErr}(\mathcal{T}_r, K)$. We prove the statement by strong induction on the height of the tree.

Fix a budget K' and let $p(\mathcal{T}_x, K')$ be the contribution to the error probability from $\mathcal{T}_x$ assuming K' sensors are placed optimally in $\mathcal{T}_x$. The base case is a subtree $\mathcal{T}_x$ of height 0, i.e., a leaf: if $K' \geq 1$ then we can place a sensor directly on the leaf and, by Lemma 3.3(iii) $p(\mathcal{T}_x, K') = 0$. If $K' = 0$ then we cannot resolve x and $p(\mathcal{T}_x, 0) = 1/N$. Now consider the general case of a rooted tree $\mathcal{T}_x$ of height $h > 0$, and assume we can find $p(\mathcal{T}_i, K'_i)$ for all trees $\mathcal{T}_i$ of height less than h. If $K' = 0$, then $p(\mathcal{T}_x, 0) = |\mathcal{T}_x|/N$ since we have no way to distinguish between any nodes in $\mathcal{T}_x$. Otherwise, we recurse over all possible partitions of K' between the subtrees rooted at the children of x (see the function $\textsc{OptErrChildren}$ in Algorithm 2). In particular, if g is the number of children of x and $\mathcal{T}_{x,i}$, for $i \in \{1, .., g\}$, denotes the subtree rooted at the i^{th} child of x, any configuration of K' sensors in $\mathcal{T}_x$ has $0 \leq K'_i \leq K'$ sensors in subtree $\mathcal{T}_{x,i}$ with $\sum_{i=1}^{g} K'_i = K'$. If $K'_i \neq K$ for every i (in particular if $K' < K$),

$$p(\mathcal{T}_x, K') = \sum_{K'_i=0} \frac{|\mathcal{T}_{x,i}|}{N} + \sum_{K'_i \neq 0} p(\mathcal{T}_{x,i}, K'_i).$$

In fact, by Lemma 3.3(i), x is equivalent to all nodes in the subtrees $\mathcal{T}_{x,i}$ (if any) for which $K'_i = 0$ and $|[x]| - 1 = \sum_{K'_i=0} |\mathcal{T}_{x,i}|$. Instead, if there exists j in $\{1, .., g\}$ such that $K'_j = K$ (all sensors are placed in the subtree $\mathcal{T}_{x,j}$),

$$p(\mathcal{T}_x, K) = \sum_{i \neq j} \frac{|\mathcal{T}_{x,i}|}{N} + p(\mathcal{T}_{x,j}, K) + \frac{1}{N},$$

because the j^{th} child of x is equivalent to x and to all nodes in the subtrees $\mathcal{T}_{x,i}$ with $i \neq j$. Since the height of each $\mathcal{T}_{x,i}$ is less than h, by the induction hypothesis we can compute the optimal $p(\mathcal{T}_{x,i}, K'_i)$, and hence $p(\mathcal{T}_x, K')$. □

Runtime Bound. A call to OptErrChildren is determined by the root x of the subtree, the subset c of its children considered and the budget $K' \leq K$. There are $N-1$ possible values for the pair (x, c) ($N-1$ is the number of edges $\mathcal{T}$). In fact, we can assume that the children are ordered and the possible partitions are of the form (c, children at the right of c)[1] so the number of pairs (x, c) is bounded by $N-1$. Hence, there are $O(NK)$ possible calls of OptErrChildren. Combining this with the minimization on $m \leq K$ sensors sent to the leftmost sub-tree, the runtime is $O(NK^2)$. □

3.4 Expected Error Distance Minimization

We now turn to the minimization of the expected error distance, which for uniform prior π, can be computed as

$$\mathcal{D}_e(\mathcal{U}) = \mathbf{E}[d(v^\star, \widehat{v})] = \frac{1}{N} \sum_{[v]_\mathcal{U} \subseteq V} \Bigg(\sum_{s,t \in [v]_\mathcal{U}} \frac{d(s,t)}{|[v]|_\mathcal{U}} \Bigg). \tag{3.4}$$

In the error-probability case, the contribution of each equivalence class was a function only of the number of its elements (see Equation (3.3)). Here instead, the contribution of each unresolved node to (3.4) depends not only on the size of the classes but also on the sum of distances between the nodes in each equivalence class; this makes the problem more challenging.

Similarly to what happens for $\mathcal{P}_e$, if $\mathcal{T}$ has ℓ leaves, placing a sensor in every leaf ensures $\mathcal{D}_e = 0$. Moreover, if $K \in \{2, .., \ell\}$ $\mathcal{U}_{\text{opt}}$ is contained in the leaf set.

Theorem 3.5. *Let $\mathcal{T}$ be a tree of maximum degree D with N nodes and ℓ leaves and let the prior π be uniform. If $K \geq \ell$, the leaves form an optimal sensor set. If $K \in \{2, .., \ell - 1\}$, there exists an algorithm that finds the set $\mathcal{U}_{\text{opt}} \in \text{argmin}_{|\mathcal{U}|=K} \mathcal{D}_e(\mathcal{U})$ in time $O(2^D N K^2)$.*

---

[1] We consider the children of any node to be ordered, e.g. according to the order induced by any embedding of $\mathcal{T}$ in the plane.

Algorithm 2 Minimizes $\mathcal{P}_e$ with budget K on a tree of size N rooted at r

```
OptErr($\mathcal{T}_x$, $K'$)
if $K' = 0$ then
    return $\frac{|\mathcal{T}_x|}{N}$
if $|\mathcal{T}_x| = 1$ then
    $p \leftarrow 0$
else
    $p \leftarrow$OptErrChildren($\mathcal{T}_x$, $K'$, children($x$))
if $x \neq r$ and $K' = K$ then
    return $p + \frac{1}{N}$
else
    return $p$

OptErrChildren($\mathcal{T}_x$, $K'$, $C$)
if $|C| = 0$ then
    return 0
else if $K' = 0$ then
    return $\sum_{c \in C} \frac{|\text{subtree}(c)|}{N}$
$f \leftarrow$ first child, $oc \leftarrow$ other children, results $\leftarrow \{\}$
for $m$ from 0 to $K'$ do
    results $\leftarrow$ results $\cup$ {OptErr($\mathcal{T}_f, m$)+ OptErrChildren($\mathcal{T}_x, K' - m, oc$)}
return min{results}
```

Correctness. Consider $\mathcal{T}$ as rooted at an arbitrary non-leaf node r. We claim that the main function of Algorithm 3, i.e., OptDist($\mathcal{T}_r, K$), finds the set $\mathcal{U}_{\text{opt}}$. The structure of Algorithm 3 is very similar to that of Algorithm 2 as is the proof of correctness for the algorithm, so we limit ourselves to highlighting the differences. When computing the expected error distance on a tree rooted at x we need to keep track of all the subtrees rooted at the children of x where there is not any sensor (see the variable *unsensored-neighbors* in the pseudo-code). Using the notation of the proof of Theorem 3.4 and applying again Lemma 3.3(i), if $K'_i \neq K$ for every subtree $\mathcal{T}_{x,i}$, the expected distance $e(T_x, K')$ of the classes entirely contained in $\mathcal{T}_x$ is computed as

$$e(\mathcal{T}_x, K') = \sum_{K'_i \neq 0} e(\mathcal{T}_{x,i}, K'_i) + \mathbf{E}[d(\widehat{v}, v^\star) | v^\star \in \{x, V(\mathcal{T}_{x,i}) : K'_i = 0\}].$$

Instead, if there exist a child x_j of x such that $K'_j = K$ (all sensors are in $\mathcal{T}_{x,j}$), $e(\mathcal{T}_{x,j}, K)$ is computed taking into account that the subtree $\mathcal{F}_{x_j}$ rooted at x_j and containing the root node r is entirely contained in the class $[x_j]$. The case $K' = 0$ never arises in the calls to OptDist since, when no sensor is assigned to a given subtree, it is enough to add its root to the list of *unsensored-neighbors* in the next calls of OptDistChildren so that the entire subtree will contribute to the final computation of the expected error distance.

Finally we look at the pre-computation of the contributions to the expected error distance. For every subtree $\mathcal{T}_x$ and subset $S = \{x_1, .., x_m\}$ of children of x for which the subtree $\mathcal{T}_{x_i}$ does not contain sensors, the expected error distance (denoted by $\textsc{ExpDist}(x, S)$ in the pseudo-code) is recursively pre-computed as follows. The base case is a subtree of only one element, i.e. a leaf, for which the expected distance is 0. For a non-leaf node x, the contribution to the expected error distance of the subtree rooted at x can be computed based on the contributions of the subtrees rooted at the children of x. We note that if $i, j \in \{1, .., m\}$, $i \neq j$,

$$\begin{aligned}\sum_{u\in\mathcal{T}_{x_i}, v\in\mathcal{T}_{x_j}} d(u,v) &= \sum_{u\in\mathcal{T}_{x_i}, v\in\mathcal{T}_{x_j}} d(u,x) + d(x,v)\\ &= |\mathcal{T}_{x_j}| \sum_{u\in\mathcal{T}_{x_i}} d(u,x) + |\mathcal{T}_{x_i}| \sum_{v\in\mathcal{T}_{x_j}} d(v,x).\end{aligned}$$

Then, if there is at least a sensor in $\mathcal{T}\backslash\mathcal{T}_x$, i.e., if $[x] \subseteq \mathcal{T}_x$, we have

$$\begin{aligned}\mathbf{E}[d(\widehat{v}, v^\star)|v^\star \in [x]] =& \sum_{j=1}^{m} \frac{|\mathcal{T}_{x_j}|}{|\mathcal{T}_x|}\mathbf{E}[d(\widehat{v}, v^\star)|v^\star \in \mathcal{T}_{x_j}]\\ &+ \frac{2}{|\mathcal{T}_x|}\sum_{j=1}^{m}\Big(|\mathcal{T}_x\backslash\mathcal{T}_{x_j}| \sum_{u\in\mathcal{T}_{x_j}} d(u,x)\Big)\\ &+ \frac{2}{|\mathcal{T}_x|}\sum_{j=1}^{m}\Big(\sum_{u\in\mathcal{T}_{x_j}} d(u,x)\Big),\end{aligned}$$

where the first term accounts for the cases in which $v^\star$ and $\widehat{v}$ are in the same subtree, the second term for the cases in which $v^\star$ and $\widehat{v}$ are in two different subtrees and the last term accounts for the cases in which either $v^\star$ or $\widehat{v}$ are in x itself. The sums of distances from x to the nodes in a given subtree that appear in the latter expression can again be recursively pre-computed. If all the available sensors are in $\mathcal{T}_x$, i.e., if $[x] \nsubseteq \mathcal{T}_x$, also the contribution to the expected error distance coming from the subtree rooted at the father of x and not containing x should be added. Once more, this contribution can be recursively pre-computed with similar techniques. □

Runtime bound. With respect to Theorem 3.4 the call to $\textsc{OptDistChildren}$ has an additional argument which corresponds to the list of neighboring subtrees that have already been considered and to which no sensor has been assigned. Since the number of neighbors of x is less than the maximum degree D, the number of possible calls to the algorithm for a given x and a sensor budget K is upper-bounded by 2^D and the total number of calls is $O(2^D NK^2)$. Finally, the expected distance for every x and all subsets of neighboring subtrees can be pre-computed with a runtime $O(2^D N)$. In conclusion the total runtime for the algorithm is $O(2^D NK^2)$. □

Note that in Algorithm 3, when $|\mathcal{T}_x| = 1$ and $K' > 1$ we return ∞: in this way the cases in which the budget is not completely allocated are directly excluded.

Algorithm 3 Minimizes the expected error distance with budget K on a tree of size N rooted at r

```
OptDist($\mathcal{T}_x$, $K'$)
if $|\mathcal{T}_x| = 1$ then
    if $K' = 1$ then return 0
    else
        return $\infty$
if $x \neq r$ and $K' = K$ then
    unsensored-neighbors ← [parent(x)]
else
    unsensored-neighbors ← [ ]
return OptDistChildren($\mathcal{T}_x$, $K$, children($x$), unsensored-neighbors)

OptDistChildren($\mathcal{T}_x$, $K'$, $C$, $S$)
if $|C|$=0 and $K' > 0$ then
    return $\infty$
if $K = 0$ then
    $S \leftarrow S \cup C$
    return ExpDist($x, S$)
if $x \neq r$ and $K' = K$ then
    for $c$ in $C$ do
        results ← {OptDist($\mathcal{T}_c$, $K$)}
$f$ ← first child, $oc$ ← other children
results ← {OptDistChildren($\mathcal{T}_x$, $K'$, $oc$, $S \cup \{f\}$)}
$h \leftarrow \min(K', K - 1)$
for $m$ from 1 to $h$ do
    $e_1$ ← OptDist($\mathcal{T}_f$, $m$)
    $e_2$ ← OptDistChildren($\mathcal{T}_x$, $K' - m$, $oc$, $N$)
    results ← $e_1 + e_2 \cup$ results
return min{results}
```

3.5 Extensions

This section presents the extension of our results to weighted nodes, i.e., to the case where choosing different sensors can result in different costs, and to general priors on the identity of the source.

3.5.1 Weighted Nodes

Theorem 3.4 and Theorem 3.5 can be extended to the more challenging case where each node u has a cost k_u and $K \in \mathbb{N}$ still represents the total budget allowed, i.e., we have the constraint $\sum_{u \in \mathcal{U}} k_u \leq K$.

The additional difficulty comes from the fact that, if each node u has a cost $k_u \in \mathbb{N}$, the optimal sensor placement $\mathcal{U}_{\text{opt}}$ will not necessarily be contained in the leaves of $\mathcal{T}$, especially if the leaves have high cost compared to other nodes of the network.

We give a few remarks on how the structure of Algorithm 2 (respectively, Algorithm 3) can be adapted to this case without increasing the total runtime bound of the algorithm. At each call of the function OptErr (respectively, OptDist) on a subtree $\mathcal{T}_x$ with budget $K' > k_x$ two possible cases need to be considered: 1) node x itself is chosen as a sensor and the remaining budget $K' - k_x$ is distributed in the subtrees rooted at the children of x; 2) node x is not chosen as a sensor and we distribute the entire budget K' among the children of x. At an algorithm level this translates in an additional call to the function OptErrChildren (respectively, OptDistChildren) for every node x. Hence the runtime bound of the algorithms is not affected.

3.5.2 Non-uniform Priors

We now consider the case of a non-uniform prior probability π on the identity of the source.

Proposition 3.6. *Let $\mathcal{G} = \mathcal{G}(V, E)$ be a network and $\mathcal{U} \subseteq V$ the set of sensors with $|\mathcal{U}| = K \geq 2$. Let $\pi(v) = \mathbf{P}(v^\star = v)$ for every $v \in V$. The error probability $\mathcal{P}_e$ and the expected distance $\mathcal{D}_e$ can be computed as follows:*

1. *Error probability*

$$\begin{aligned}\mathcal{P}_e = \mathbf{P}(v^\star \neq \widehat{v}) &= \sum_{[v]_{\mathcal{U}} \subseteq V} \left(\sum_{s,t \in [v]_{\mathcal{U}}, s \neq t} \frac{\pi(s)\pi(t)}{\pi([v]_{\mathcal{U}})} \right) \\ &= \sum_{[v]_{\mathcal{U}} \subseteq V} \left(\sum_{s \in [v]_{\mathcal{U}}} \frac{\pi(s)(\pi([v]_{\mathcal{U}}) - \pi(s))}{\pi([v]_{\mathcal{U}})} \right).\end{aligned} \tag{3.5}$$

2. *Expected error distance $\widehat{v}$*

$$\mathcal{D}_e = \mathbf{E}[d(v^\star, \widehat{v})] = \sum_{[v]_{\mathcal{U}} \subseteq V} \left(\sum_{s,t \in [v]_{\mathcal{U}}} \frac{d(s,t)\pi(s)\pi(t)}{\pi([v]_{\mathcal{U}})} \right). \tag{3.6}$$

In both Equation (3.5) and Equation (3.6) the contribution to each equivalence class depends on the prior probability of each element in the class and, in the case of Equation

(3.6), on the distances between elements in a same class. If we now think of the recursive computation of $\mathcal{P}_e$ and $\mathcal{D}_e$ in a tree, in both cases, the contribution of a node u depends on which of the subtrees rooted at the children of u contain or do not contain sensors and not only on their cardinality. Hence, in view of Theorem 3.5 and Algorithm 3, it is clear that for both metrics it is possible to find an optimal set $\mathcal{U}_{\text{opt}}$ in time $O(2^D N K^2)$. The proof of the result follows that of Theorem 3.5.

3.6 Discussion

In this chapter, we presented optimal algorithms for placing a limited budget of sensors to be used for source localization. Our algorithms build on the notion of node resolvability, which we showed to be the key to studying optimal sensor placement for source localization. We considered two metrics of practical relevance, error probability and expected error distance, and we gave a polynomial time solution for optimizing them on trees. We showed that our approach is quite general, comprising interesting extensions such as to sensors with different costs and to general priors on the identity of the source.

An interesting extension would study *worst-case* metrics rather than *average-case* metrics; e.g., minimizing the maximum distance between a real and estimated source could be of interest.

The results we presented in this chapter shed light on the importance of sensor placement for source localization and on the advantages of studying sensor placement in relation to double resolvability. However, these results are limited by two strong assumptions: the deterministic spread of the epidemics and the absence of cycles in the network. In many practical situations, we are uncertain about the transmission delays or the transmission delays are subject to high randomness, in which cases the optimality of our sensor placements is not guaranteed. Finally, it could be thought possible to use the algorithms we derived on a general network using a tree-approximation of the network. Unfortunately, this is not the case. In fact, the sensor set obtained would highly depend on the particular approximating tree. Moreover, different approximating trees would lead to sensor sets that can reliably identify different set of possible sources: The distances from a node to the sensors in the approximating subtree would typically be close to the real distances if the node is in the core of the tree but this would not be the case if the node is a leaf.

In the next chapter, we develop an approach to sensor placement that does not suffer from the mentioned restrictions.

4 Sensor Placement on General Networks: The Effect of the Transmission-Delays Variance

In the previous chapter we have studied sensor placement for source localization under two strong assumptions: a tree topology and low-variance transmission delays. Here, we study sensor placement with a more general approach that does not rely on any of these two assumptions.

The results of Chapter 3 heavily-rely on the tree-structure on the network, and they cannot be directly extended to general networks. However, the concept of node-resolvability, which we introduced in Chapter 3, will again be crucial for our analysis. As for the variance of the transmission delays, we will show that this model parameter has a strong impact on source localization and, in particular, on the optimal sensor placement.

Our main results in this chapter are two distinct algorithms for sensor placement that are specifically designed for two different regimes of transmission delays: a low-variance regime and a high-variance regime. In fact, as illustrated in Figure 4.1 and 4.2, as the variance of the transmission delays grows, the optimal sensor set must change in order to fit the growing difficulty of source localization.

These results were published in [Spinelli et al., 2016, 2017a].

4.1 Overview

We study the choice of sensors for source localization and, in particular, how it is affected by the *amount* of randomness (i.e., the variance) of the transmission delays $\{X_{uv}\}_{uv \in E}$. To this end, we separately analyze two different regimes for the amount of randomness of the transmission delays: *low-variance* and *high-variance.* A dichotomy exists between the two, and our approach for sensor placement differs. We take a principled approach that begins with considering deterministic transmission delays (*zero-variance* regime), and we build on this intuition in order to develop heuristics for both *low-variance* and

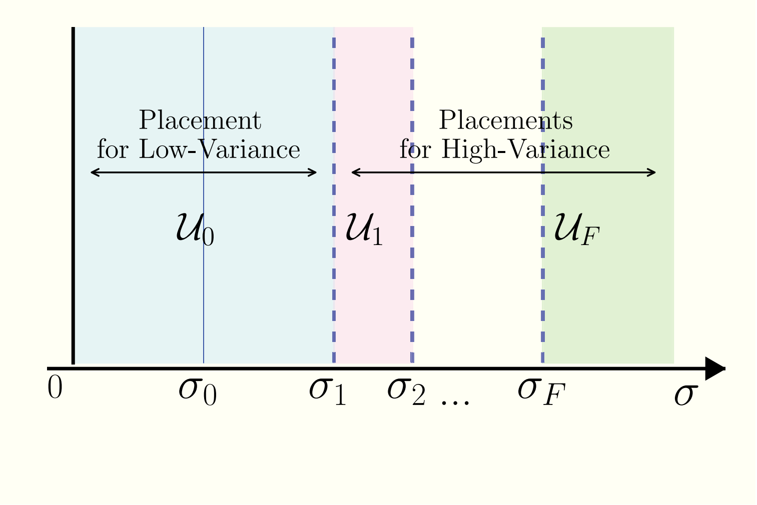
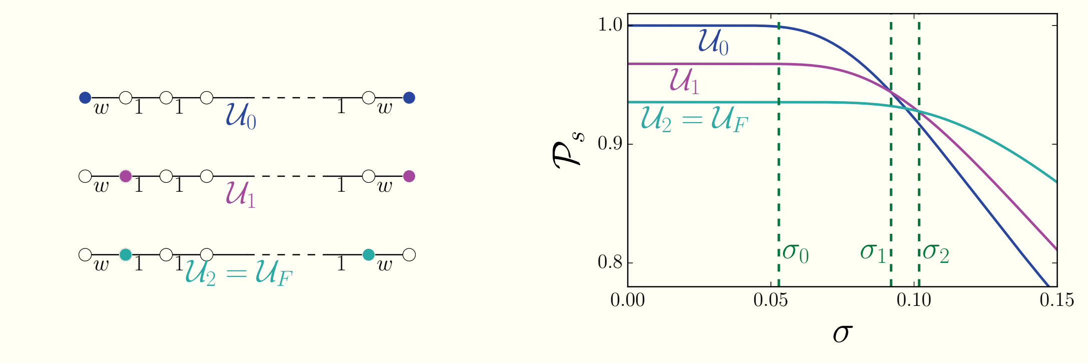


Figure 4.1 – Sequence of optimal sensor placements for increasing transmission variance. We assume the transmission transmission delays $\{X_{uv}\}_{uv \in E}$ to be such that $\mathbf{E}[X_{uv}] = w_{uv} \in \mathbb{R}_+$ and such that the variance is a growing function of a *variance parameter* σ, i.e., $\mathrm{Var}(X_{uv}) = g(w_{uv}, \sigma)$ with $g(x, 0) = 0$ for all $x \in \mathbb{R}^+$. For $\sigma \in (0, \sigma_0)$ the transmission delays are effectively deterministic (i.e., σ does not affect source localization). For $\sigma \in (\sigma_0, \sigma_1)$, σ affects the accuracy of source localization but the optimal sensor placement is still $\mathcal{U}_0$. For larger σ, the optimal sensor placement might change, possibly multiple times ($\mathcal{U}_k$ denotes the optimal placement for $\sigma \in (\sigma_k, \sigma_{k+1})$) up to $\sigma = \sigma_F$. For $\sigma > \sigma_F$ the optimal placement remains the same ($\mathcal{U}_F$).

high-variance regimes.

Low-Variance Regime

When the variance in the transmission delays is *low* (see Section 4.3), we prove that the set of optimal sensors is exactly the optimal set for the zero-variance regime. In the zero- and low- variance regime, both the probability of success $\mathcal{P}_s$ (as well as other possible metrics of interest) can be explicitly computed. Despite this seeming simplicity, the problem remains NP-hard (see Section 1.2.4). As in Chapter 3, we tackle the problem by using its connection with the well-studied double-resolving-set (DRS) problem [Cáceres et al., 2007] that minimizes the number of sensors with which the source can be localized. This minimum number is, in many cases, still prohibitively large, hence we cannot use this approach directly. However, from the connection between sensor placement and DRS, we find inspiration for our algorithm which, by selecting one sensor at a time in order to reach a DRS set until the budget is exhausted, greedily improves $\mathcal{P}_s$ (see Section 4.3).



(a) (b)

Figure 4.2 – Optimal sensors for Gaussian-distributed transmission delays with unit mean and standard deviation σ on a path network. In this case $\mathcal{P}_s$ and, consequently, the optimal sensor placements, can be explicitly computed. **(a)**: different sensor placements; **(b)**: their performance in terms of probability of success $\mathcal{P}_s$ for $w = 20$ and 30 edges.

High-Variance Regime

When the noise in the transmission delays is *high*, it is no longer negligible and it poses an additional challenge to source localization; in effect, the accumulation of noise from node to node as the epidemic spreads might no longer enable us to distinguish between two potential sources, especially when they are both *far* from all sensors. Hence, we must *strengthen* the requirements for sensor placement in order to ensure that the nodes can be distinguished by sensors that are *near* to them; this nearness is a function of the noise, of the budget K, and of the network topology. We define a novel objective function that both maximizes the success probability and imposes a *uniform* spread of sensors in the network. Taking inspiration from the low-variance regime, we design an algorithm that greedily maximizes this new objective (see Section 4.4).

Empirical Results

In Section 4.5, we evaluate our algorithms on three different real-world datasets that represent different application areas for source localization and different network topologies. First, we take a community of people living in the proximity of a university campus [Aharony et al., 2011], a typical network for the transmission of airborne diseases. Second, we take a community of students exchanging messages over a Facebook-like social network [Opsahl and Panzarasa, 2009] through which ideas and trends can propagate. Finally, we consider the road network of the state of California [Census]: This captures geographical networks that can model the transmission of a disease between connected communities or the diffusion of contaminants, e.g., through a hydrological network. We show that our methods perform favourably against state-of-the-art approaches in both the low- and the high-variance regimes yielding an improvement of up to 50%. (see Section 4.5.2). Moreover, in the empirical results, the dichotomy between the low- and high-variance regimes becomes apparent.

4.2 Preliminaries

4.2.1 Model

We assume the epidemic model described in Section 1.2. Recall that the *transmission delay* through edge uv, i.e., the time it takes for a node u to infect a neighbor node v is encoded by the random variable X_{uv}. We consider a transmission model which is both natural and versatile as it comprises deterministic transmissions, which we call *zero-variance*, and arbitrary *random* independent transmission models. A large part of the epidemic literature models transmission delays with exponential random variables. However we make a different modeling choice for two reasons. First, we are interested in decoupling the transmission variance and the average transmission time (for exponential random variables, mean and variance cannot be tuned independently). Second, in many applications it has been suggested that the transmission delays can be less-skewed than exponential random variables [Cha et al., 2009, Lessler et al., 2009, Vergu et al., 2010]. For every edge uv we assume X_{uv} to be a symmetric and non-negative[1] random variable. We do not make any strong assumption on the distribution of the transmission delays X_{uv}: we only assume that their mean is equal to the edge weights, i.e., $\mathbf{E}[X_{uv}] = w_{uv}$ for every $uv \in E$, and that their variance is an increasing function of both the edge weight and of a variance parameter σ, that is, $\mathrm{Var}(X_{uv}) = g(w_{uv}, \sigma)$, where g depends on the particular distribution of X_{uv} and $g(x, 0) = 0$ for all $x \in \mathbb{R}^+$.

If the variance parameter is zero, or if it is low compared to the edge weights, network distances are a good proxy for time delays (see Section 1.2.2 and Section 4.3 here). We refer to this setting as a *low-variance* regime, as opposed to the *high-variance* regime in which time delays are very noisy and network distances no longer work as a proxy for time delays.

4.2.2 Source Localization

Let $\mathcal{U} \subseteq V$ be the set of sensors (which we will select). We assume we know the time at which each sensor is infected, and we refer to this vector of infection times as $T_{\mathcal{U}}$. Knowing $T_{\mathcal{U}}$ is a standard and realistic assumption [Netrapalli and Sanghavi, 2012]. We want to identify the source using only the information contained in $T_{\mathcal{U}}$. We use maximum

---

[1] Note that in Figures 4.2 and 4.3a we compute the value of the success probability $\mathcal{P}_s$ assuming Gaussian distributed delays (and ignoring that, with low probability, negative delays could appear) because this is the only distribution that makes the exact computation of this value feasible. However, in all experiments we only consider non-negative distributions for X_{uv}.

likelihood estimation (MLE) to produce an estimate $\widehat{v}$ of the true unknown source $v^\star$ as in Pinto et al. [2012]. This approach is common (see e.g., Shah and Zaman [2011], Dong et al. [2013]), although the exact form of the estimator depends on the model and assumptions. In our case we have

$$\widehat{v} \in \operatorname{argmax}_{s \in V} \mathbf{P}(T_{\mathcal{U}} | v^\star = s) \pi(v^\star),$$

where π denotes the prior on the identity of the source. In this chapter we assume π to be uniform (i.e., $\pi(v) = 1/N$ for all nodes $v \in V$).

4.2.3 Metrics

We assume that we are given a *budget* K on the number of sensors we can use, and that we must select our sensors *once and for all*, i.e., independently of any particular epidemic instance. In order to select the *best set of sensors* $\mathcal{U}$ *of size* K we must first define our metric of interest. In this chapter we are mainly interested in the *success probability*

$$\mathcal{P}_s = \mathbf{P}(\widehat{v} = v^\star)$$

which is a widely used metric for source localization (see, e.g., Shah and Zaman [2011], Pinto et al. [2012], Louni and Subbalakshmi [2014]). In our experiments we also evaluate another important metric, the *expected distance* between the estimated source and the real source, i.e., $\mathcal{D}_e = \mathbf{E}[d(v^\star, \widehat{v})]$, where d denotes the distance between two nodes in the network.

4.3 The Low-variance Regime

In the zero-variance setting, i.e., when $X_{uv} = w_{uv}$ for every edge $uv \in E$ the equivalence class of the source can be identified based on the observation vector $\mathbf{t}_{u_1}$ as described in Section 3.2.

As in Chapter 3, if a prior distribution π on the identity of the source is given, after the class of the source is identified we can generate an estimated source $\widehat{v}$ sampling from $\pi|_{[v^\star]}(u) = \mathbf{P}(v^\star = v | v \in [v^\star])$. If a prior π is not known, we sample $\widehat{v}$ uniformly at random from $[v^\star]$, which is equivalent to having a uniform prior π.

Actually, we can estimate the source with this simple algorithm also in a more general *low-variance* setting. In fact, as mentioned in Chapter 3, if the transmission delays are random but do not deviate much from their average values we are always able to identify the equivalence class to which the real source belongs by finding the class $[v]$ that minimizes the distance between the distance vector $\mathbf{d}_{v,u_1}$ and the observation vector $\mathbf{t}_{u_1}$.

4.3.1 Noise Tolerance

Proposition 4.1 extends Proposition 3.1 to the case of general networks proving a looser version of the bound on tolerated-noise of Section 3.2.

Proposition 4.1. *Let* $\mathcal{G}(V, E)$ *be a network of size* N*,* $\mathcal{U} \subseteq V$ *and fix* $u_1 \in \mathcal{U}$*. Call*

$$\phi \triangleq \min_{v,z:\mathbf{d}_{v,u_1} \neq \mathbf{d}_{z,u_1}} \|\mathbf{d}_{v,u_1} - \mathbf{d}_{z,u_1}\|_\infty$$

and call Δ *the maximum distance in any shortest path between any node and any sensor.*

If the transmission delays are such that for each $uv \in E$*,* $X_{uv} \in [w_{uv}(1-\varepsilon), w_{uv}(1+\varepsilon)]$ *with* $\varepsilon < \varepsilon_0 \triangleq \phi/(4\Delta)$*, the equivalence class of the source is* $[v^\star] = [v]$ *such that*

$$[v] = \arg\min_{z \in V} \|\mathbf{d}_{z,u_1} - \mathbf{t}_{u_1}\|_\infty.$$

Proof. Let $t_{u'}$ be the infection time of $u' \in \mathcal{U}$. When the source is $v^\star$ we have

$$t_{u'} - t^\star \leq d(v^\star, u')(1+\varepsilon). \tag{4.1}$$

Moreover, if $\mathcal{Q}$ is the collection of all paths connecting $v^\star$ and u' and, for $p \in \mathcal{Q}$, if $d_p(v^\star, u')$ is the (weighted) length of path p we have

$$t_{u'} - t^\star \geq \min_{p \in \mathcal{Q}} d_p(v^\star, u')(1-\varepsilon) = d(v^\star, u')(1-\varepsilon). \tag{4.2}$$

Combining inequalities (4.1) and (4.2) for u' being, respectively, u and u_1 and calling t_1 (resp., t_u) the infection time of the reference sensor u_1 (resp., u), we have

$$|t_u - t_1 - d(v^\star, u) + d(v^\star, u_1)| \leq \varepsilon(d(v^\star, u) + d(v^\star, u_1)) \leq 2\varepsilon\Delta.$$

Since for every $v \in [v^\star]$ $\mathbf{d}_{v,u_1} = \mathbf{d}_{v^\star,u_1}$, we conclude that for every $v \in [v^\star]$, $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 2\varepsilon\Delta$.

Take now $v \notin [v^\star]$ and assume by contradiction that $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 2\varepsilon\Delta$. Using the triangular inequality and the hypothesis $\varepsilon < \phi/4\Delta$ we have

$$\|\mathbf{d}_{v^\star,u_1} - \mathbf{d}_{v,u_1}\|_\infty \leq \|\mathbf{d}_{v^\star,u_1} - \mathbf{t}_{u_1}\|_\infty + \|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty \leq 4\varepsilon\Delta < \phi,$$

which contradicts the definition of ϕ. Hence for every $v \notin [v^\star]$, $\|\mathbf{d}_{v,u_1} - \mathbf{t}_{u_1}\|_\infty > 2\varepsilon\Delta$. □

Note that here ε_0 plays the role of σ_0 in Figure 4.1 in the sense that it is an upper-bound on a regime in which the delays are effectively deterministic and the variance of the transmission delays does not affect the accuracy of source localization.

For the remainder of this section, we will assume $\varepsilon < \phi/4\Delta$, which we call the low-variance regime.

4.3.2 Algorithm for Sensor Placement

Applying Proposition 4.1, the success probability $\mathcal{P}_s$, as well as other possible metrics of interest, can be computed exactly in polynomial time independently of the topology of the network $\mathcal{G}$ (see, e.g., Equation (1.3) and (1.4) in Chapter 1). In fact, due to Lemma 1.9 and Theorem 4.1, it is enough to compute the distance vector of Definition 1.3 for all the nodes. Nonetheless, if we have a budget $K \geq 2$ of nodes that we can choose as sensors, finding the configuration that maximizes $\mathcal{P}_s$ is an NP-hard problem (see Section 1.2.4).

For trees, the optimal sensor placement can be find in polynomial time using dynamic programming techniques (see Chapter 3). In a general network (with loops) the problem of source localization is made more challenging by the multiplicity of paths through which the epidemic can spread and for the same reason also finding an optimal sensor set becomes much harder.

A first idea to solve sensor placement on a general network could be to use the latter result on a breadth-first-search (BFS) approximation of the network. However, as mentioned in Section 1.2.4, on a tree the optimal sensor placement is contained in the leaf set. If we consider a non-tree network and take a BFS-approximation, the leaves of the BFS tree depend on where the BFS-tree is rooted. Hence, by applying the result of Chapter 3 to a tree approximation, it is not possible to guarantee high probability of success independently of the identity of the source.

Our approach, presented in Algorithm 4, does not rely on a network approximation. Moreover, it is specifically designed for the source localization problem and has a simple greedy structure: for every node $v \in V$, initialize $\mathcal{U} \leftarrow \{v\}$ and iteratively add to $\mathcal{U}$ the node u that maximizes the gain with respect to the success probability until we either run out of budget or $\mathcal{P}_s = 1$. Equation (1.3) ensures that greedily maximizing $\mathcal{P}_s$ is equivalent to greedily maximizing the number q of equivalence classes. When adding an element to the sensor set, the partition in equivalence classes can be updated in linear time, hence the total runtime of our algorithm is $O(KN^3)$. Despite bypassing the NP-hardness of the problem, this might not be sufficiently fast for very large networks. However, the algorithm is extremely parallelizable (see, for example, the main **for** loop and the **argmax** in the **while** loop).

The sensor placement obtained through Algorithm 4 will be denoted LV-Obs to emphasize the fact that it is designed for the case in which the variance is absent or very small (LV stands for low-variance regime).

Unfortunately we cannot use a submodularity argument to give guarantees on the

Algorithm 4 (LV-Obs): sensor placement for the low-variance setting.

```
Require: Network $\mathcal{G}$, budget $K$
  for $v \in V$ do
      $\mathcal{U}_v \leftarrow v$
      while $\mathcal{P}_s(\mathcal{U}_v) \neq 1$ and $\mathcal{U}_v < K$ do
          $u \leftarrow \mathrm{argmax}_{z \in V \setminus \mathcal{U}_v} [\mathcal{P}_s(\mathcal{U}_v \cup \{z\}) - \mathcal{P}_s(\mathcal{U}_v)]$
          $\mathcal{U}_v \leftarrow \mathcal{U}_v \cup \{u\}$.
  return $\mathrm{argmax}_{v \in V} \mathcal{P}_s(\mathcal{U}_v)$
```

performance of Algorithm 4 because the number of equivalent classes, and hence the function $\mathcal{P}_s$, are not submodular. Consider as a simple example a cycle $\mathcal{C}$ of length 6 as in Figure 4.4a. If the sensor set is $\mathcal{U}_1 = \{1\}$, the number of equivalence classes is $q = 1$. If we add node 2 to $\mathcal{U}_1$ the classes become $\{1, 5, 6\}$ and $\{2, 3, 4\}$ $(q = 2)$. Hence by adding node 2 to the set $\{1\}$ the gain in terms of equivalence classes is just 1. Consider now $\mathcal{U}_2 = \{1, 4\} \supseteq \mathcal{U}_1$, which identifies as classes $\{1\}$, $\{4\}$, $\{2, 6\}$ and $\{3, 5\}$. If again we add node 2 to $\mathcal{U}_2$ we reach a DRS of $\mathcal{C}$, i.e., all classes are singletons. This means that the gain in terms of equivalence classes is $6 - 4 = 2 > 1$ and we conclude that the number of equivalence classes is not submodular.

4.3.3 The Approximating Algorithm of [Chen et al., 2014]

Chen et al. [2014] proposed a $(1 + o(1)) \log(N)$-approximation algorithm to find the minimal size of a DRS, i.e., the double metric dimension (DMD) defined in Section 1.2.2. Their algorithm is based on a minimization of an *entropy* function[2] and shares with LV-Obs the property of being greedy. The entropy minimized by Chen et al. [2014] is defined as follows.

Definition 4.2 (Entropy of Chen et al. [2014]). *Let $\mathcal{G}$ a network, $\mathcal{U} \subseteq V$, a set of sensors. The entropy of $\mathcal{U}$ is*

$$H_{\mathcal{U}} = \log_2 \left(\prod_{[u]_{\mathcal{U}} \subseteq V} |[u]_{\mathcal{U}}|! \right)$$

Note that $H_{\mathcal{U}}$ is minimized if and only if each equivalence class consists of only one node and hence if and only if $\mathcal{P}_s = 1$ (see Section 1.2.3).

However, despite the fact that $H_{\mathcal{U}}$ is minimized when $\mathcal{P}_s$ is maximized and that both act on the same set of equivalence classes for a given $\mathcal{U}$, the greedy processes that minimize $H_{\mathcal{U}}$ and maximize $\mathcal{P}_s$ are not the same. This can be seen by rewriting both objective functions in the following way. Let $c_1, \ldots, c_q$ be the sizes of the equivalence classes. Then

---

[2] Note that this has no connection to the information-theoretic entropy

$H_{\mathcal{U}}$ can be written as

$$H_{\mathcal{U}}(c_1, .., c_q) = \sum_{i=1}^{l} \sum_{j=2}^{c_i} \log(j) = \sum_{i=2}^{\max c_j} \log(i) \cdot |\{c_j \geq i\}|.$$

Analogously we have the following equality for the success probability

$$\mathcal{P}_s(c_1, \ldots, c_q) = N(1 - \mathcal{P}_s(c_1, \ldots, c_q)) = N - q = \sum_{i=2}^{\max c_j} |\{c_j \geq i\}|.$$

Hence, though similar in spirit, a greedy minimization of $H_{\mathcal{U}}$ is not related to a greedy maximization of $\mathcal{P}_s$.

4.3.4 Comparison with Benchmarks

As budgeted sensor placement (even in the zero-variance setting) is NP-hard, there is no optimal algorithm to compare against. Instead, we evaluate the performance of our algorithm against a set of natural benchmarks that have shown to have good performance in other works [Seo et al., 2012, Berry et al., 2006, Zhang et al., 2016] (see Section 4.5.2 for a discussion of these benchmarks, Figure 4.8-4.10 for the results).

We further compare LV-Obs against two other natural heuristics that also optimize an objective function greedily.

Alternative Objective Functions

The first baseline heuristic we consider is an adapted version of the approximation algorithm for the DRS problem proposed by Chen et al. [2014] and described in Section 4.3.3. By stopping the greedy process after it selects K nodes, we can adapt in a natural way this approximation algorithm and create a heuristic for the budgeted version that we denote by Φ_{ent}. We want to check if LV-Obs actually reaches smaller values of $\mathcal{P}_s$ compared to Φ_{ent}.

The second is a greedy minimization of the expected error distance $\mathcal{D}_e = \mathbf{E}[d(v^\star, \widehat{v})]$ of Equation (1.4) that we denote by Φ_{dist}. Even if LV-Obs is not directly minimizing $\mathcal{D}_e$, we want to compare the results we obtain in terms of $\mathcal{D}_e$ with those obtained by Φ_{dist} in order to check if, at least in some budget regimes, we can use the maximization of $\mathcal{P}_s$ as a proxy for the minimization of $\mathcal{D}_e$.

Let us here denote LV-Obs with Φ for consistency of notation. Table 4.1 compares LV-Obs, Φ_{ent} and Φ_{dist}, for different topologies and different budgets k, in terms of both $\mathcal{P}_s$ and $\mathcal{D}_e$. The results are given in the form of (averaged) relative differences.We denote

Random Geometric Network, $N = 100$, $r = 0.2$

	$\rho(\mathcal{P}_s, \Phi, \Phi_{dist})$	$\rho(\mathcal{D}_e, \Phi_{dist}, \Phi)$	$\rho(\mathcal{P}_s, \Phi, \Phi_{ent})$
$k = 2$	-0.205	0.101	-0.033
$k = 4$	-0.014	-0.003	-0.007
$k = 8$	-0.003	-0.002	-0.003

Barabási Albert Network, $N = 100$, $m = 3$

	$\rho(\mathcal{P}_s, \Phi, \Phi_{dist})$	$\rho(\mathcal{D}_e, \Phi_{dist}, \Phi)$	$\rho(\mathcal{P}_s, \Phi, \Phi_{ent})$
$k = 2$	-0.168	0.023	-0.037
$k = 4$	-0.039	0.025	-0.028
$k = 8$	-0.004	-0.003	0.005

Table 4.1 – Comparison of LV-Obs (Φ) with the greedy algorithms that minimize the entropy function of Chen et al. [2014] (Φ_{ent}) and the expected distance (Φ_{dist})

the relative difference of x and y with respect to f as

$$\rho(f, x, y) \overset{\text{def}}{=} \frac{f(y) - f(x)}{f(x)}.$$

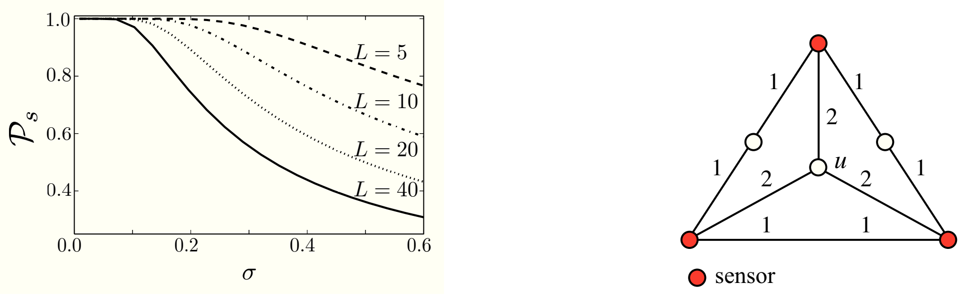
Since the expected distance can be equal to 0 we add 1 to the denominator when comparing values of $\mathcal{D}_e$, i.e.,

$$\rho(\mathcal{D}_e, x, y) \overset{\text{def}}{=} \frac{\mathcal{D}_e(y) - \mathcal{D}_e(x)}{\mathcal{D}_e(x) + 1}.$$

The results achieved by Φ_{ent} and Φ_{dist} are, on average, worse than those of Algorithm 4 both in terms of $\mathcal{P}_s$ and of $\mathcal{D}_e$, independently of the network topology. We observe two exceptions. First, when k is very small: Φ_{dist} reaches smaller values of $\mathcal{D}_e$ compared to LV-Obs, which can be explained by the fact that Φ_{dist} directly minimizes $\mathcal{D}_e$ and that, when fewer sensors are available the difference between the sensor placements that maximize $\mathcal{P}_s$ and minimize $\mathcal{D}_e$ is greater. Second, for large k, on the Barabási Albert networks Φ_{ent} gives, in average, larger $\mathcal{P}_s$ than LV-Obs. This is probably due to the fact that, for this class of networks, the DMD is small, hence with a large value of k we approach the regime in which the objective function of Φ_{ent}, designed to minimize the DMD of the network, is optimal.

4.4 The High-variance Regime

When the variance is not guaranteed to be low, as defined in Section 4.3, computing analytically the success probability - or other metrics of interest - is unfortunately not possible (except for very simple networks, like the path network of Figure 4.2, and for



(a) (b)

Figure 4.3 – **(a)**: Success probability $\mathcal{P}_s$ on a path of length L for increasing variance σ. **(b)**: Counterexample for the converse of Lemma 4.3; for each pair of sensors in $\mathcal{U}$, u is not contained in the shortest path between them, yet $\mathcal{U}$ is a DRS.

particular transmission delays, e.g., Gaussian-distributed).

When the variance is high, also the localization of the source is more challenging because the observed infection delays $t_i - t_j$ can be misleading, especially if the corresponding sensors u_i and u_j are *far* from the source. Take, for example, a path of length L where the two leaves are the only two sensors and all edges have weight equal to 1. Figure 4.3a shows how the success probability $\mathcal{P}_s$ decays faster for increasing values of L. Building on this observation, we propose a strategy for sensor placement that enforces a controlled distance from a general source node to the sensor set.

4.4.1 Estimation of the Source

For the high-variance case we localize the source using an adapted version of the algorithm proposed by Pinto et al. [2012]. This adapted algorithm can be seen as a generalization to the high-variance regime of the source localization method used for the low-variance regime.

Denote by $T_{\mathcal{U}}$ the vector of the observed infection times. If the transmission delays are Gaussian-distributed, $\mathcal{G}$ is a tree, the maximum likelihood (ML) estimator defined as

$$\widehat{v} \in \arg\max_{v \in V} \mathbf{P}(v | T_{\mathcal{U}}),$$

has a tractable closed form [Pinto et al., 2012]. Note that the model of Pinto et al. [2012] additionally assumes that the infected sensors reveal which neighbor infected them; this assumption is not essential for the derivation of the ML estimator and it is not required in our work.

In particular, given a set of sensors

$$\mathcal{U} = \{u_1, u_2, \ldots, u_K\} \subseteq V,$$

and assuming that, $X_{uv} \sim \mathcal{N}(w_{uv}, \sigma w_{uv})$, the vector of the observed infection delays $\mathbf{t}_{u_1} = [t_2 - t_1, \ldots, t_K - t_1] \in \mathbb{R}^{K-1}$ is distributed as $\mathcal{N}(\mathbf{d}_{v,u_1}, \mathbf{\Lambda}_{\mathcal{U}})$ where $\mathbf{d}_{v,u_1}$ is the distance vector of Definition 1.8 and the covariance matrix $\mathbf{\Lambda}_{u_1}$ is

$$\mathbf{\Lambda}_{u_1,(k,i)} = \sigma^2 \begin{cases} \sum_{(u,v)\in\mathcal{P}(u_1,u_{k+1})} w_{uv}^2 & k = i \\ \sum_{(u,v)\in\mathcal{P}(u_1,u_{k+1})\cap\mathcal{P}(u_1,u_{i+1})} w_{uv}^2 & k \neq i. \end{cases} \tag{4.3}$$

Hence the ML estimator is

$$\begin{aligned} \widehat{v} &\in \arg\max_{v\in V} \frac{\exp\Big(-\frac{1}{2}(\mathbf{t}_{u_1} - \mathbf{d}_{v,u_1})^\top {\mathbf{\Lambda}_{u_1}}^{-1} (\mathbf{t}_{u_1} - \mathbf{d}_{v,u_1})\Big)}{|\mathbf{\Lambda}_{u_1}|^{1/2}} \\ &= \arg\max_{v\in V} \Big[\mathbf{d}_{v,u_1}^\top {\mathbf{\Lambda}_{u_1}}^{-1} (\mathbf{t}_{u_1} - \frac{1}{2}\mathbf{d}_{v,u_1})\Big]. \end{aligned} \tag{4.4}$$

On non-tree networks, the multiplicity of paths linking any two nodes makes source estimation more challenging. As claimed in [Pinto et al., 2012], the same estimator can be used as an approximation of the ML estimator for a non-tree network by assuming that the diffusion happens only through breadth-first-search (BFS) tree rooted at the (unknown) source. In this case the paths which appear in the definition of the covariance matrix $\mathbf{\Lambda}_{u_1}$ are computed on the BFS tree rooted at the candidate source considered. Hence $\mathbf{\Lambda}_{u_1}$ depends on the candidate source and the ML estimator is

$$\widehat{v}_{\text{BFS}} \in \arg\max_{v\in V} \frac{\exp\Big(-\frac{1}{2}(\mathbf{t}_{u_1} - \mathbf{d}_{v,u_1})^\top {\mathbf{\Lambda}_{u_1}^{v}}^{-1} (\mathbf{t}_{u_1} - \mathbf{d}_{v,u_1})\Big)}{|\mathbf{\Lambda}_{u_1}^{v}|^{1/2}}. \tag{4.5}$$

In this work, we adopt (4.5) as the source estimator in the noisy case. In fact, even we do not assume transmission delays to be Gaussian-distributed, under the hypothesis of sparse observations, we can apply the Central Limit Theorem (CLT) to approximate the sum of the edge delays with Gaussian random variables: if all edges have the same weight we can apply the CLT for i.i.d. random variables; if this is not the case, we can apply Lyapunov's version of the CLT.[3] Finally, using (4.5) to compute the ML estimator, the likelihood of nodes in the same equivalence class can result to be different as an artefact of the BFS approximation. Hence, for consistency with our source localization method in

---

[3] Lyapunov condition with $\delta = 1$ is easily verified for a sequence of independent and uniformly bounded random variables (see Example 27.4 in [Billingsley, 1995] for more details).

the low-variance case, we compute an average likelihood and estimate that the source is in the class with the higher average likelihood. Then, once an equivalence class for the source is estimated, we select $\widehat{v}$ by sampling the prior probability on the identity of the source (if available), or by uniform sampling, from the estimated equivalence class.

4.4.2 Algorithm for Sensor Placement

First, we formalize why distances between sensors are important. Recall that for every transmission delay X_{uv} we assume $\mathrm{Var}(X_{uv}) = g(w_{uv}, \sigma)$, with g being an increasing function of both its arguments. If u_i, u_j are two sensors connected by a unique path $\mathcal{P}(u_i, u_j)$ and the source is $v^\star \in \mathcal{P}(u_i, u_j)$, then

$$\mathrm{Var}(t_i - t_j) = \Bigg[\sum_{uv \in \mathcal{P}(u_i,u_j)} g(w_{uv}, \sigma). \Bigg]. \tag{4.6}$$

For example, if $X_{uv} \sim \mathcal{N}(w_{uv}, \sigma^2 w_{uv}^2)$ we have

$$\mathrm{Var}(t_i - t_j) = \sigma^2 \Bigg[\sum_{uv \in \mathcal{P}(u_i,u_j)} w_{uv}^2 \Bigg]. \tag{4.7}$$

Although we cannot control σ, we can control the *path length* between sensors.

We make use of the following sufficient condition for a set to be a DRS, i.e., for a sensor set to guarantee correct source localization.

Lemma 4.3. *Let $\mathcal{G}(V, E)$ be a network, $\mathcal{U} \subseteq V$. If for every $u \in V$ there exist $u_1, u_2 \in \mathcal{U}$ such that there is a unique shortest path $\mathcal{P}(u_1, u_2)$ between u_1 and u_2 and $u \in \mathcal{P}(u_1, u_2)$, then $\mathcal{U}$ is a* DRS *for $\mathcal{G}$.*

Proof. Let $u, v \in V \backslash \mathcal{U}$. We will prove that there exist $u_1, u_2 \in \mathcal{U}$ such that the pair (u, v) is resolved by (u_1, u_2), i.e., $d(v, u_1) - d(u, u_1) \neq d(v, u_2) - d(u, u_2)$. Let $u_1, u_2 \in \mathcal{U}$ such that u appears in the unique shortest path $\mathcal{P}(u_1, u_2)$ and $u_3, u_4 \in S$ such that v appears in the unique shortest path $\mathcal{P}(u_3, u_4)$. If $v \in \mathcal{P}(u_1, u_2)$ or $u \in \mathcal{P}(u_3, u_4)$ than u and v are resolved by, respectively, (u_1, u_2) or (u_3, u_4). Take $v \notin \mathcal{P}(u_1, u_2)$ and $u \notin \mathcal{P}(u_3, u_4)$. In this case, $\{u_1, u_2\} \neq \{u_3, u_4\}$. Let us suppose without loss of generality that $u_1 \notin \{u_3, u_4\}$. We look only at the case where (u_1, u_2) does not resolve (u, v) and prove that the pair is indeed resolved by two nodes in $\mathcal{U}$. Since (u_1, u_2) does not resolve (u, v), there exists $c \in \mathbb{R}$ such that $d(v, u_1) - d(u, u_1) = c = d(v, u_2) - d(u, u_2)$. Since the unique shortest path between u_1 and u_2 goes through u we have that $c > 0$. We prove that either (u_1, u_3) or (u_1, u_4) resolves (u, v). If this was not the case, we would have the following equalities:

$$\begin{aligned} c &= d(v, u_1) - d(u, u_1) = d(v, u_3) - d(u, u_3) \\ c &= d(v, u_1) - d(u, u_1) = d(v, u_4) - d(u, u_4). \end{aligned}$$

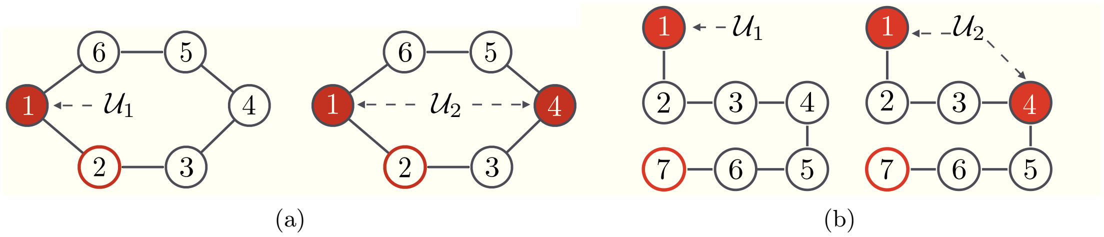


Figure 4.4 – Counterexamples for the submodularity property of Algorithm 1 (a) and Algorithm 2 (b). For the network in (a) (respectively, (b)) the gain of adding the node with red border to $\mathcal{U}_2$ is larger in terms of $\mathcal{P}_s$ (respectively, P_L) than the gain of adding it to $\mathcal{U}_1 \subseteq \mathcal{U}_2$.

Since $c > 0$, $d(v, u_3) > d(u, u_3)$ and $d(v, u_4) > d(u, u_4)$ giving a contradiction with v (and not u) being on the shortest path $\mathcal{P}(u_3, u_4)$. We conclude that (u, v) are resolved by either (u_1, u_3) or (u_1, u_4). □

The converse of this lemma is not true: If $\mathcal{U}$ double resolves $\mathcal{G}$, it is not even true that for every node u there must exist $u_1, u_2 \in \mathcal{U}$ such that u is contained in *some* shortest path between u_1 and u_2 of (see Figure 4.3b).

Path Covering Strategy

We take Lemma 4.3 as a basis for deriving a *path covering* strategy for sensor placement. In practice, the condition about the *uniqueness* of the shortest path is too strong and excludes many potentially useful sensors. Experimentally we see that in many practical situations two shortest paths differ only by a few nodes and the majority of nodes on the path are resolved by the two extreme nodes. This is why we relax the condition of Lemma 4.3 and we prefer, when the shortest path is not unique, to select one arbitrarily. Let $\mathcal{U} \subseteq V$ be a set of sensors and L a positive integer: We call $P_L(\mathcal{U})$ the set of nodes that lie on a shortest path of length at most L between any two sensors in the set $\mathcal{U}$. Given a budget K, and a positive integer L, we denote by $\mathcal{U}^{\star}_{K,L}$ the set of K nodes that maximize the cardinality of $P_L(\mathcal{U})$. We call L the *length constraint* for the sensor placement because we consider an sensor to be *useful* for source localization only if it is within distance L from another sensor. $\mathcal{U}^{\star}_{K,L}$ can be approximated greedily as in Algorithm 5. The runtime of Algorithm 5 is $O(N^2K^2)$, however, as Algorithm 4, this algorithm is highly parallelizable and hence tractable even for large networks.

We will refer to the sensor placement produced by Algorithm 5 as HV-Obs(L) to emphasize that it is designed for the high-variance case.

Unfortunately also for Algorithm 5 we cannot use a submodularity argument to derive approximation guarantees. In fact, the function P_L is not submodular. Consider the path

Algorithm 5 (HV-Obs): sensor placement for the high-variance setting.

```
Require: Network $\mathcal{G}(V,E)$, budget $K$, length constraint $L$
  $N \leftarrow |\mathcal{G}|$
  for $v \in V$ do
    $\mathcal{U}_v \leftarrow v$
    while $|P_L(\mathcal{U}_v)| \neq N$ and $\mathcal{U}_v < K$ do
      $u \leftarrow \text{argmax}_{z \in V \setminus \mathcal{U}_v} [|P_L(\mathcal{U}_v \cup \{z\})| - |P_L(\mathcal{U}_v)|]$
      $\mathcal{U}_v \leftarrow \mathcal{U}_v \cup \{u\}$.
  return $\text{argmax}_{v \in V} |P_L(\mathcal{U}_v)|$
```

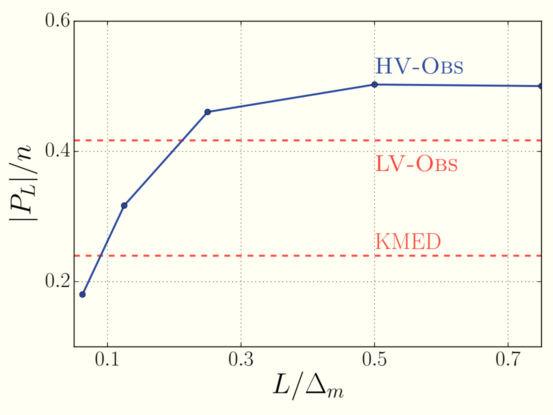
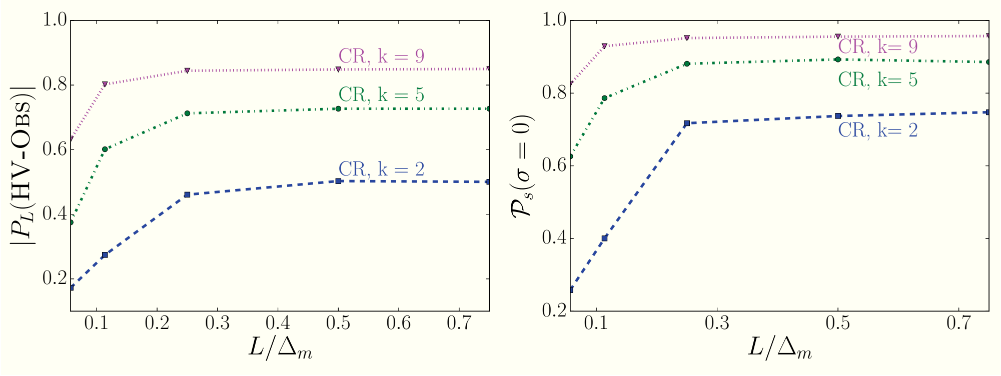


Figure 4.5 – Fraction of nodes in $P_L(\cdot)$ for the CR dataset with 2% of sensors.

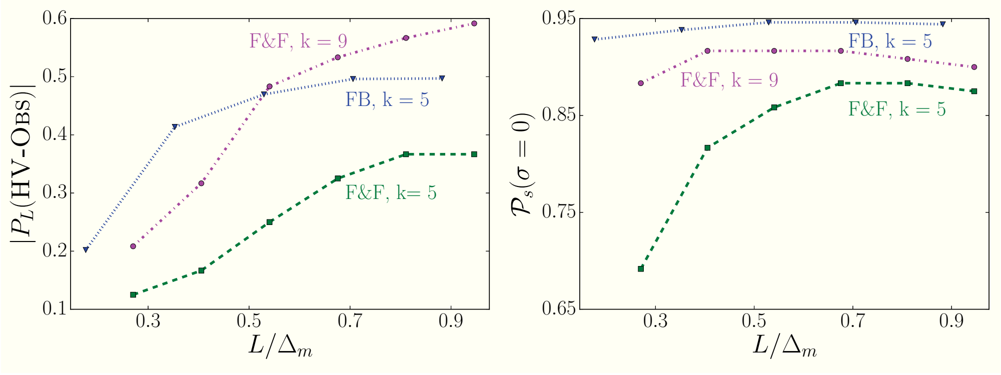
$\mathcal{P}$ of 7 nodes in Figure 4.4b, fix $L = 3$ and set $\mathcal{U}_1 = \{1\}$. If we add node 7 to $\mathcal{U}_1$ no node lies on a path of length smaller than $L = 3$ among the two sensors 1 and 7, hence the gain is 0. Consider now $\mathcal{U}_2 = \{1, 4\} \supseteq \mathcal{U}_1$. If we add node 7 to $\mathcal{U}_2$, the gain is 3 because node $5, 6$ and 7, that did not lie on any path of length smaller than L connecting two sensors before, now lie on the path connecting 4 and 7, hence P_L is not submodular.

Comparison with Algorithm 4

Note that taking L equal to the maximum weighted distance Δ_m between two nodes in $\mathcal{G}$ does not make Algorithm 5 equivalent to Algorithm 4, i.e., we do not obtain LV-Obs. To see how the two algorithms could give different results, take a cycle of odd length d with a leaf ℓ added as a neighbor to an arbitrary node v and assume to start the algorithm with initial set $\{v\}$. At the first step, the two algorithms will make the same choice, choosing one of the two nodes that are at distance $(d-1)/2$ from v. At the second step however, LV-Obs will add ℓ (by Corollary 1.7 a DRS contains all leaves), whereas Algorithm 5 will add a node on the cycle. This observation is key to our results because it explains why Algorithm 5 results in a more uniform (and hence *variance-resistant*) sensor placement with respect to LV-Obs. HV-Obs operates a trade-off between the average distance to



(a) CR (b) CR



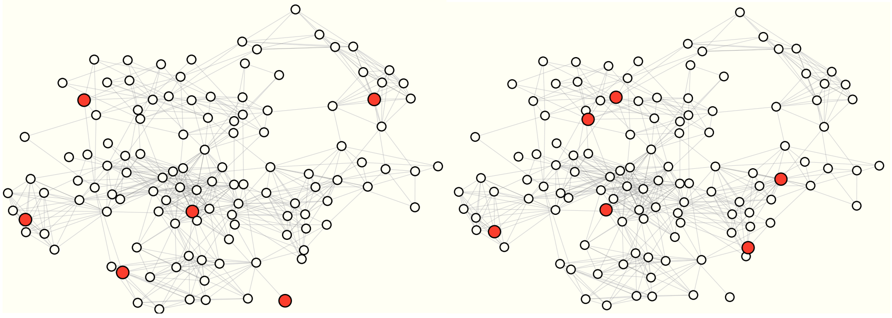
(c) F&F, FB (d) F&F, FB

Figure 4.6 – Fraction of nodes in $P_L(\text{HV-Obs})$ and success probability in the zero-variance regime ($\mathcal{P}_s(\sigma = 0)$) as a function of L/Δ_m.

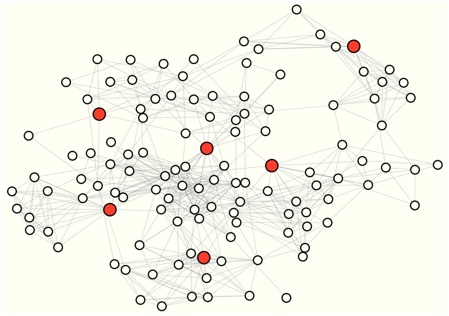
the sensors and the maximization of $\mathcal{P}_s$.

Choice of the L parameter

How could one optimally set L? Needless to say, the optimal L depends on the network topology and on the available budget: Clearly, for a larger budget a smaller L is preferred. The cardinality of $P_L(\mathcal{U})$ is a good proxy for the performance of $\mathcal{U}$. The value $|P_L|$ is increasing in L and reaches its maximum for L equal to the maximum weighted distance Δ_m. For small L, $|P_L(\text{HV-Obs})| < |P_{\Delta_m}(\text{LV-Obs})|$ but for L large enough this is no longer the case. See Figure 4.5 for an example. Our empirical results suggest that L should be chosen as the maximum for which $|P_L(\text{HV-Obs})| \leq |P_{\Delta_m}(\text{LV-Obs})|$. The key property of HV-Obs with respect to LV-Obs is that sensors are spread more *uniformly* without *losing* too much in terms of success probability $\mathcal{P}_s$: Figure 4.6 shows $|P_L(\text{HV-Obs})|$ and $\mathcal{P}_s$ as a function of L. An a-priori evaluation of the variance threshold above which one should use the HV-Obs placement (and of the appropriate value of the L parameter) can be based on the comparison of $\mathcal{P}_s$ on a path network for different



(a) LV-Obs (b) HV-Obs, $\frac{L}{\Delta_m} = 0.5$



(c) KMED

Figure 4.7 – Comparison of the oberver placements LV-Obs, HV-Obs ($\frac{L}{\Delta_m} = 0.5$) and KMED with $K = 5\%$ on the F&F network. Note the difference between LV-Obs and HV-Obs: LV-Obs contains leaves while HV-Obs has shorter spacing.

values of L and σ as in Figure4.3a. In fact, looking at Figure4.3a we see that, for small values of σ $\mathcal{P}_s$ is very close to 1 independently of L, hence LV-Obs is the best solution. When σ grows, we see that, in order to guarantee an high $\mathcal{P}_s$ one must choose smaller and smaller values of L. LV-Obs and HV-Obs can give drastically different sensors (see Figure 4.7 for an example).

4.5 Experimental Results

4.5.1 Datasets

We purposely run our experiments on three very different real-world networks that, in addition to being relevant examples of networks for epidemic spread, display different characteristics in terms of size, diameter, clustering coefficient and average degree (see Table 4.2), enabling us to test the performance of our methods on various topologies. The three networks we consider are:

◇ Friend & Families (F&F). This is a dataset containing phone calls, SMS exchanges

	$\lvert V\rvert$	$\lvert E\rvert$	$\min(w_{uv})$	$\text{avg}(w_{uv})$	$\max(w_{uv})$
Friends & Families (F&F)	120	563	4	5.58	7
Facebook Messages (FB)	1020	6205	1	2.97	5
California Roads (CR)	1259	1801	1	1.71	9

	Avg Degree	*Diameter*	*Avg Distance*	*Avg Clustering*
Friends & Families (F&F)	9.38	6	17.5	0.67
Facebook Messages (FB)	12.16	5	6.69	0.09
California Roads (CR)	2.86	66	55.3	0.2

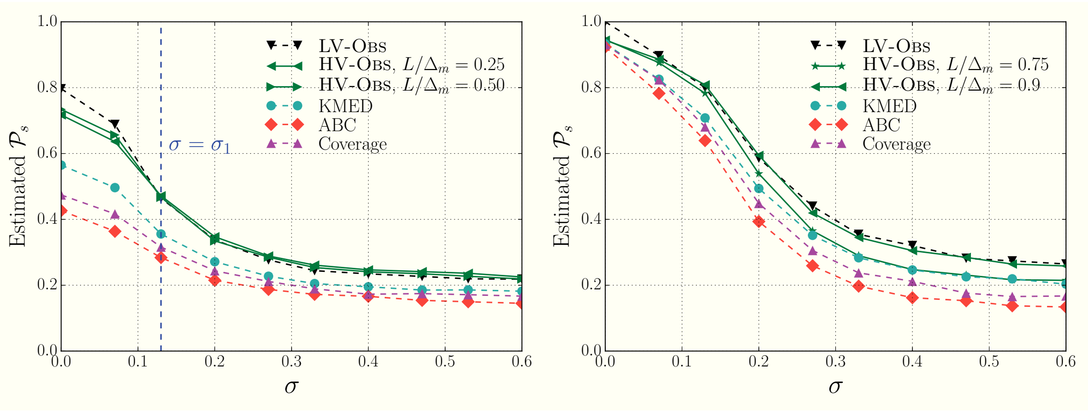
Table 4.2 – Statistics for the networks examined

and bluetooth proximity, among a community living in the proximity of a university campus [Aharony et al., 2011]. We select the largest connected component of individuals who took part in the experiment during its whole duration. The edges are weighted, according to the number of phone calls, SMSs, and bluetooth contacts.

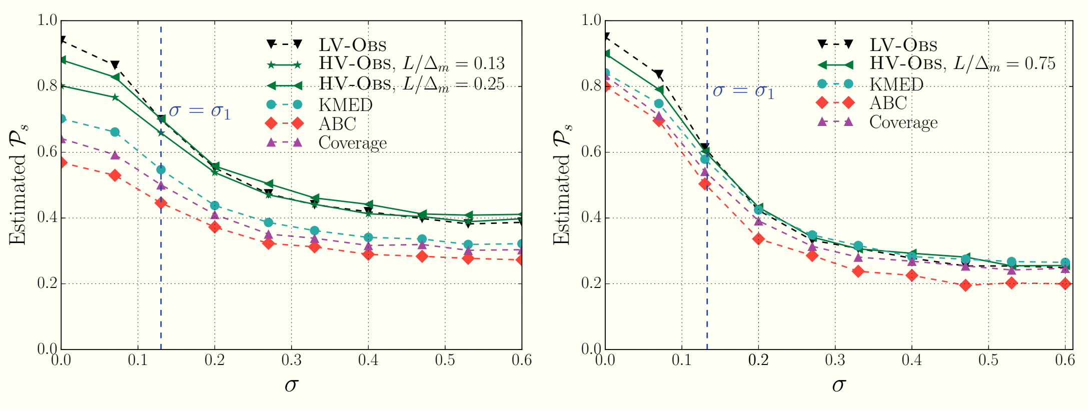
◇ Facebook-like Message Exchange (FB) [Opsahl and Panzarasa, 2009]. As the individuals included in this dataset were living on the same university-campus, the number of messages exchanged is likely to be a good measure of in-person interaction. We selected links on which at least one message was sent in both directions and individuals that had a contact with at least one other individual.

◇ California Road Network (CR) [Census]. In order to obtain a single connected component and remove points that effectively represent the same location, we collapsed the points falling within a distance of 2 km. Moreover we iteratively deleted all leaves. In fact, the roads that cross the state border are not completely tracked in this dataset and terminate with a leaf. Some other leaves might represent remote locations, not necessarily close to the borders, but their influence on the epidemic should anyway be very low. The diameter of the CR network is very large compared with that of the other two networks. The edges are weighted according to a rescaled version of the real distance (measured in km).

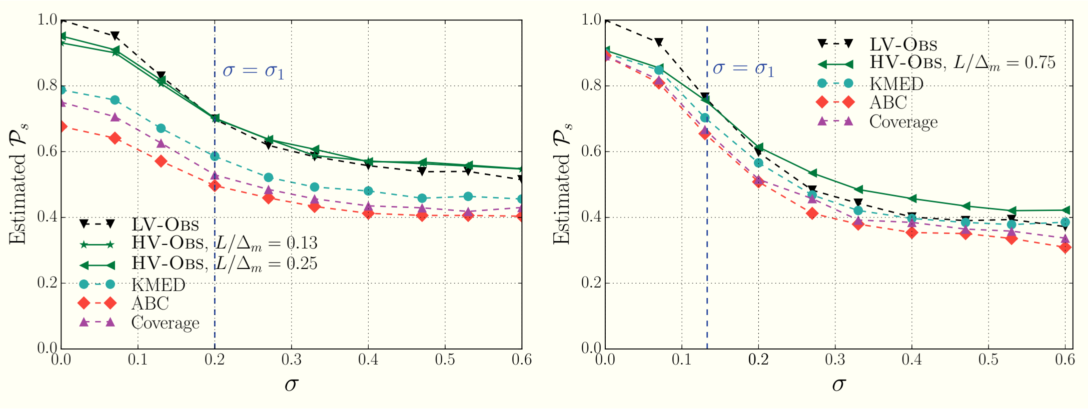
In all three networks, edges are given (non-unit) integer *weights*, which is realistic in many applications as the expected transmission delays are known only up to some level of precision. Integer weights do *not* simplify the localization of the source; in fact, this makes it *more* difficult to distinguish between nodes. For example, if the edges of the CR network were weighted according to the Euclidean distance between the two endpoints, LV-Obs would use only a very small portion of the budget and the comparison with other sensor placements would not be meaningful.



(a) CR, 2% sensors (b) FB, 5% sensors



(c) CR, 5% sensors (d) F&F, 5% sensors



(e) CR, 9% sensors (f) F&F, 10% sensors

Figure 4.8 – Success probability $\mathcal{P}_s$ as the variance parameter σ increases.

4.5.2 Comparison with Benchmarks

We compare LV-Obs and HV-Obs against the following benchmarks:

- ◇ ABC (Adaptive Betweenness Centrality): Betweenness Centrality (BC) is a popular method for placing sensors for source localization (see, e.g., Louni and Subbalakshmi [2014] and Seo et al. [2012], where it emerges as the best heuristic for sensor placement among those tested). It consists of the K nodes having the largest BC, which is defined, for all $u \in V$ as

  $$\mathrm{BC}(u) = \sum_{x,y \in V, x \neq y} \frac{\psi_{x,y}(u)}{\psi_{x,y}}$$

  where $\psi_{x,y}$ is the number of shortest paths between x and y and $\psi_{x,y}(u)$ is the number of those paths that passes through u. Here we consider an adaptive version of BC (ABC) which iteratively chooses the node that maximizes the betweenness centrality without considering the shortest paths that pass by already-chosen nodes [Yoshida, 2014]. ABC, with respect to the basic BC, gives less clustered, and hence more efficient, sensor sets.

- ◇ Coverage [Zhang et al., 2016]: This approach maximizes the number of nodes that have a sensor as neighbor, i.e.,

  $$\mathrm{C}(\mathcal{U}) = \frac{\left| \bigcup_{u \in \mathcal{U}} N_u \right|}{N}$$

  where N_u denotes the set of neighbors of u. It has been shown to outperform several heuristics with a diffusion model and a source localization setting that are very similar to ours [Zhang et al., 2016].

- ◇ K-Medians (KMED): this is the optimal placement for the closely-related problem of maximizing the detectability of a flow [Berry et al., 2006]. The KMED placement is the set of K nodes $\mathcal{U}$ such that

  $$\mathcal{U} = \arg \min_{|\mathcal{U}|=K} \sum_{v \in V} \min_{u \in \mathcal{U}} d(v,u).$$

  Determining the KMED of a network is NP-hard [Kariv and Hakimi, 1979], hence we approximate KMED with a greedy heuristic.

4.5.3 Distributions for the Transmission Delays

Unless otherwise specified, we sample the transmission delays X_{uv} from truncated Gaussian random variables with parameters $\left(w_{uv}, \sigma w_{uv}, \left[\frac{w_{uv}}{2}, \frac{3w_{uv}}{2}\right]\right)$. More precisely, if $Y_{uv} \sim$

$\mathcal{N}(w_{uv}, \sigma w_{uv})$ is a Gaussian random variable, X_{uv} is obtained by conditioning Y_{uv} with $Y_{uv} \in \left[\frac{w_{uv}}{2}, \frac{3w_{uv}}{2}\right]$. With respect to the delay distribution assumed by Pinto et al. [2012] i.e., $X_{uv} \sim \mathcal{N}(w_{uv}, \sigma w_{uv})$, the distribution we assume has the advantage of admitting only strictly positive transmission delays. Furthermore, different values of the parameter σ result in different regimes for the transmission delays, making our model very versatile. When $\sigma = 0$, we are in the zero-variance regime; when σ is large, the distribution of X_{uv} becomes closer to a uniform random variable $U\left(\left[\frac{w_{uv}}{2}, \frac{3w_{uv}}{2}\right]\right)$. Finally, when σ is strictly positive but small, $X_{uv} \approx \mathcal{N}(w_{uv}, (\sigma w_{uv})^2)$.

To assess the robustness of our approach for source localization and sensor placement, we also experiment with uniformly distributed transmission delays, i.e., for every edge $uv \in E$, we take $X_{uv} \sim \text{Unif}([(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}])$. The uniform distribution is, among the unimodal distributions on a bounded support, the one that maximizes the variance [Gray and Odell, 1967]. Hence, uniform delays are a very challenging setting for source localization.

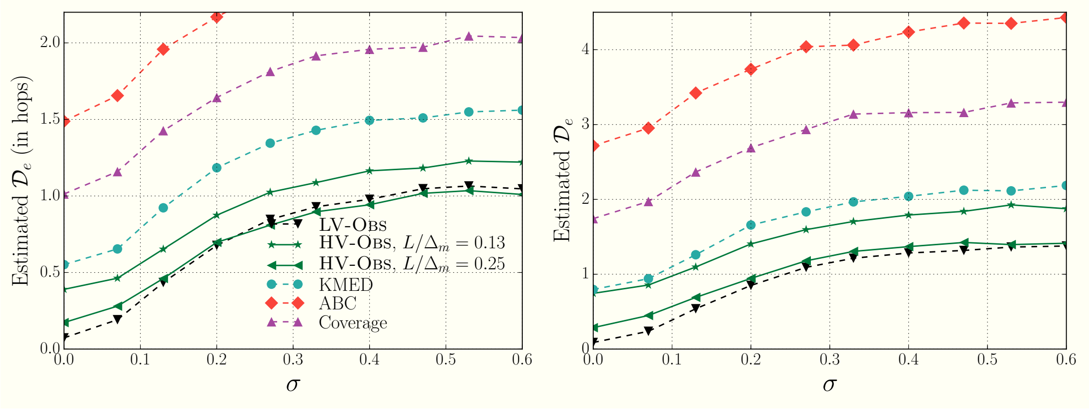
4.5.4 Evaluation of the Probability of Success and of the Expected Error Distance

We estimate the probability of success $\mathcal{P}_s$ and the expected error distance $\mathcal{D}_e$ for different values of the variance parameter σ. Our estimations are computed averaging the results obtained by choosing each node in turn as the source and generating synthetic epidemics. For the FB and the CR datasets, we run 5 simulations per node and value of σ; for the F&F dataset, as the network is smaller, we run 20 simulations per node and value of σ. For the FB and CR datasets, we localize the source based on the first 20 observations only: Given the large size of these networks, it would be unrealistic to wait for all the nodes to get infected before running the algorithm.

The results for $\mathcal{P}_s$ are displayed in Figure 4.8. An approximation of the value σ_1, above which HV-Obs outperforms LV-Obs, is marked with a vertical line. For the expected distance (weighted and in hops), see Figure 4.9.

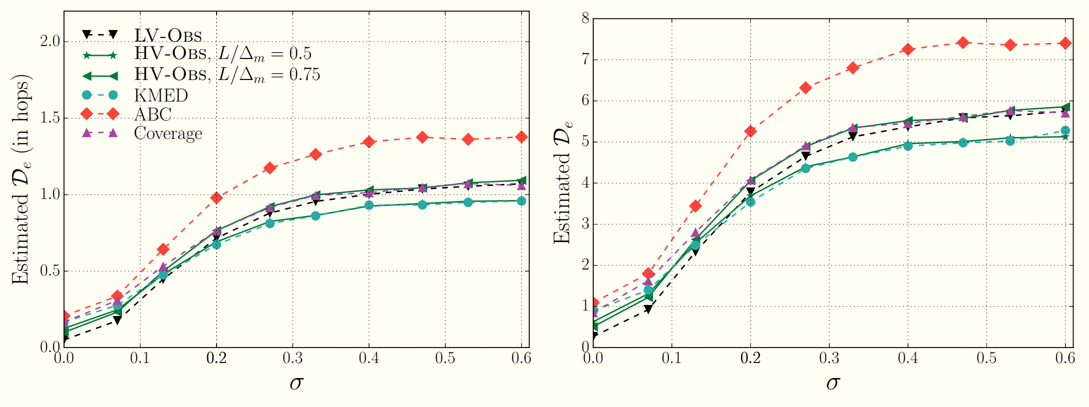
We first take as budget for the sensors the minimum budget for which $\mathcal{P}_s(\text{LV-Obs}) = 1$. This corresponds to $K \sim 10\%$ for the F&F dataset, $K \sim 9\%$ for the CR network and $K \sim 5\%$ for the FB dataset. This is the setting in which we expect the improvement of HV-Obs over LV-Obs to be especially strong: For smaller values of K we expect LV-Obs to be nearly optimal even in the high-variance regime because we do not have enough budget to contrast both the topological *undistinguishability* among nodes (what LV-Obs is designed for) and the accumulation of variance (what HV-Obs is designed for).

For the F&F and the CR networks, we also experiment with smaller percentages of sensors



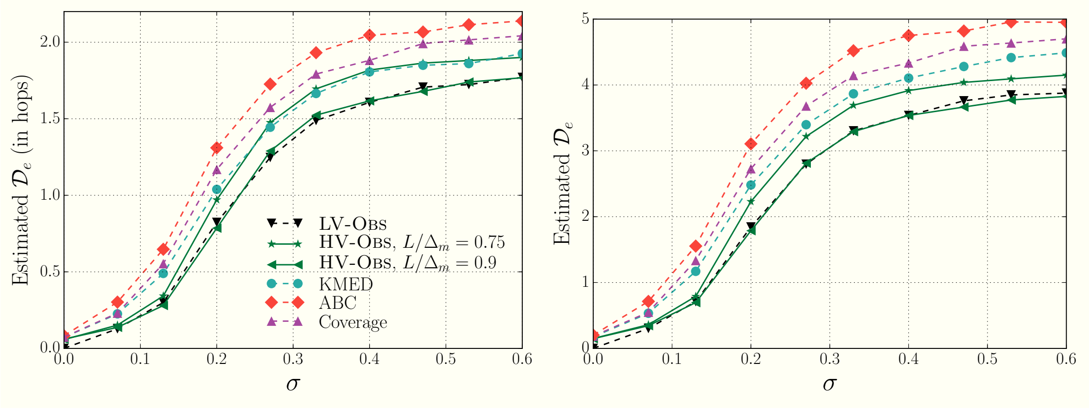
(a) CR, 5% sensors

(b) CR, 5% sensors



(c) F&F, 5% sensors

(d) F&F, 5% sensors



(e) FB, 5% sensors

(f) FB, 5% sensors

Figure 4.9 – Expected error distance $\mathcal{D}_e = \mathbf{E}[d(v^\star, \widehat{v})]$ in number of edges (first row) and in weighted path length (second row) as the variance parameter σ increases. For the plots on the right-hand side, see the plots on the left-hand side for the legend.

and consistently find an improvement of HV-Obs over LV-Obs in the high-variance regime: Below a certain amount of variance σ_1, LV-Obs performs better than HV-Obs for any choice of the parameter L, whereas above σ_1 a calibrated choice of L leads to a significant improvement. Such L stays constant for all $\sigma > \sigma_1$, i.e., with the notation of Figure 4.1, we have $\sigma_1 = \sigma_F$.

Instead, for the FB dataset, probably due to the low diameter with respect to the number of nodes, we observe that HV-Obs does not improve on LV-Obs for any value of L.

Both LV-Obs and HV-Obs systematically outperform the baseline heuristics that we described in Section 4.5.2. For the CR dataset the performance of ABC is particularly poor. The Coverage heuristic outperforms ABC on all three networks (confirming the findings of Zhang et al. [2016]) but is consistently less effective than KMED and our methods.

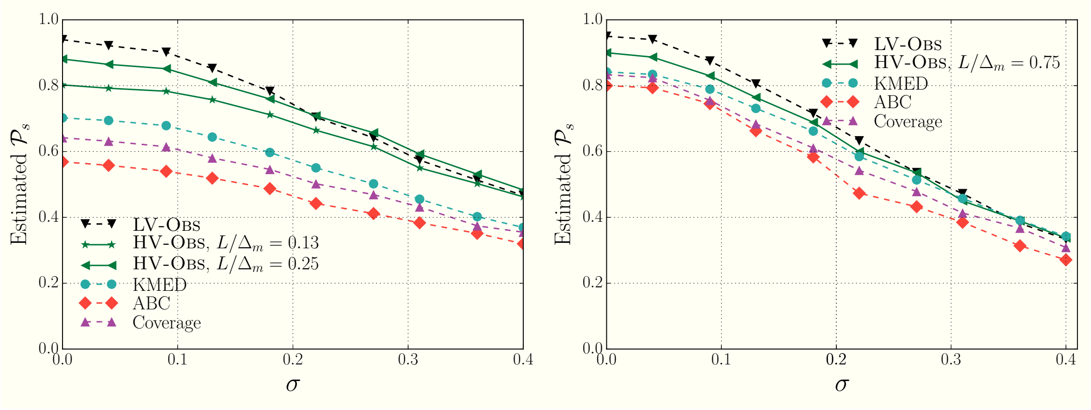
Finally in Figure 4.10, we consider uniform transmission delays, and we measure whether, without making any changes, our sensor placement still performs well. We find comparable results, which suggest that our sensor placement is not dependant on the exact transmission model and that the variance of the transmission delays is really a key factor for a good sensor placement.

4.6 Discussion

In this chapter, we presented a principled approach towards budgeted sensor placement for source localization, which shows a dichotomy between the low- and high-variance regimes. We developed complementary approaches to handle both of these regimes. Moreover, we evaluated our approaches against state-of-the-art and alternative heuristics, showing a better performance of the algorithms proposed.

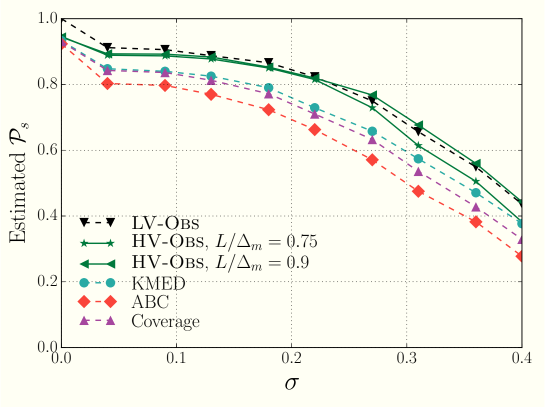
A question that remains open concerns the possibility of deriving analytical results for optimal sensor placement as a function of the transmission variance. These would certainly require adopting some additional assumptions. For example, in a similar model, under the assumption that the budget for sensors is large enough, Zejnilović et al. [2015b] recently proposed an analytical approach to sensor placement. It would be interesting to investigate if their approach extends to our setting. Alternatively, we could consider only acyclic networks and work, for example, with exponential transmission delays within the dynamic-message-passing framework of Lokhov et al. [2014].

A different, yet related, line of work would be that of a *repeated game* where the source is placed by an adversary who knows which nodes are sensors. Although largely not true for virulent epidemics, this could be a factor in the case of biowarfare or virtual attacks. Some studies, similar in spirit to this line of work, look at how to place airport security checkpoints in order to catch attackers [Pita et al., 2011] or, from the adversary point of



(a) CR, 5% sensors

(b) F&F, 5% sensors



(c) FB, 5% sensors

Figure 4.10 – Success probability $\mathcal{P}_s$ for uniform transmission delays $X_{uv} \sim \text{Unif}([(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}])$.

view, at introducing delays in the diffusion in order to better obfuscate the source [Fanti et al., 2015].

In the next chapter we present a natural extension of the work we discussed in this chapter, i.e., the case in which the observation occurs in two stages. In the first stage, as in this chapter, a small set of sensors are selected to monitor the network. In the next stage, once an epidemic begins, additional sensors are deployed in the relevant region of the network to localize the source. As we will see, this adaptive strategy yields enormous savings in terms of the number of sensors required to localize the source.

5 Sensor Placement on General Networks: Static Vs Dynamic

In Chapters 3 and 4 we have studied source localization algorithms that use the information provided by a fixed set of K sensors chosen independently of any particular epidemic. Intuitively, however, depending on the identity of the source, the most informative sensors are not always the same. Hence, for most epidemics, a large fraction of the deployed sensors are, in effect, useless. What if we could deploy a part of our sensor budget adaptively, i.e., depending on the particular epidemic instance?

Together with limiting the cost of data collection, another important concern for source localization is timeliness: If an epidemic is detected while it is spreading, being able to promptly identify its source based on the incomplete information available can be essential for the activation of containment measures [Al Qathrady et al., 2016]. Therefore, we are interested in a model where the source can be localized progressively by incorporating in the estimation all the information available, as soon as it becomes available.

In this chapter, we make minimal assumptions on the epidemic spread and design a flexible framework for information collection and source localization where the information can be either collected adaptively or at a fixed set of locations, and the source of an epidemic can be promptly localized. This framework incorporates the settings of Chapter 3 and 4, where information collection was static, i.e., not adaptive. Furthermore the results of Chapter 4 concerning the dependence of sensor placement on the transmission variance will be shown to efficiently guide the choice of sensors even in this more general setting.

These results were partly presented in [Spinelli et al., 2017b]; a more complete version was recently published in [Spinelli et al., 2018].

5.1 Overview

We localize the source by using the information provided by a subset of nodes called sensors. 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 possible types of sensors: *static* sensors and

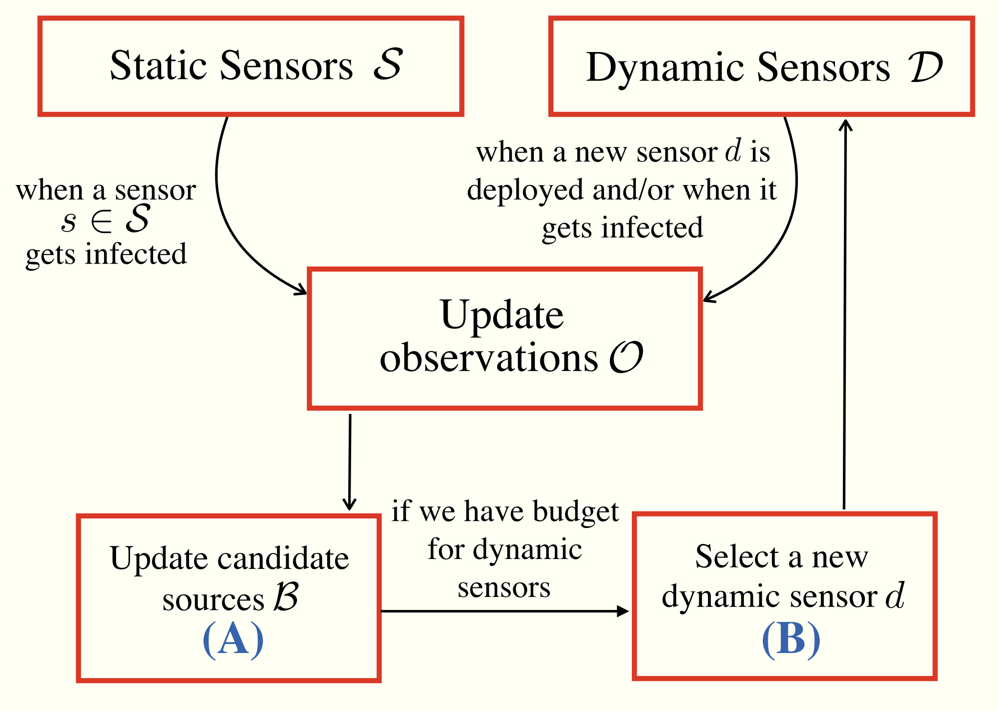

Static Sensors $\mathcal{S}$
Dynamic Sensors $\mathcal{D}$
when a sensor $s \in \mathcal{S}$ gets infected
when a new sensor d is deployed and/or when it gets infected
Update observations $\mathcal{O}$
Update candidate sources $\mathcal{B}$
(A)
if we have budget for dynamic sensors
Select a new dynamic sensor d
(B)

Figure 5.1 – Illustration of our approach to source localization. An epidemic is observed through the sensors (static $\mathcal{S}$ and dynamic $\mathcal{D}$). Based on the observations $\mathcal{O}$, we iteratively update a set $\mathcal{B}$ of candidate sources (step A) which is used, if the budget allows it, to guide the choice of an additional dynamic sensor (step B). The subroutines (A) and (B) depend on the setting used (among the four listed in Table 5.1).

dynamic sensors. Static sensors are placed *a priori* in the network, independently of any particular epidemic instance. Dynamic sensors are placed *adaptively* while we perform source localization. Figure 5.1 depicts our approach to source localization.

We propose a general framework for source localization that encompasses both *static* and *dynamic* sensor placement and that allows to localize the source both while the epidemic is still spreading (*online* localization) and after the epidemic has spread throughout the entire network (*offline* localization); see Table 5.1. This opens new possibilities, as to date most approaches assume that all sensors are static and that the source can be localized only after the epidemic spreads throughout the network. We show that when we sequentially deploy dynamic sensors, the source is always correctly identified with only a few sensors even when the transmission delays are highly noisy. This result is very practical because it applies to general networks.

We also propose several methods for choosing where to deploy the dynamic sensors and we compare them.

Because of its flexibility, the proposed framework can be used in a number of different

	Offline Localization	*Online Localization*
Static sensor placement	**S-OFF – Section 5.3** only static sensors, the source is localized after the epidemic, observations are always positive.	**S-ON – Section 5.4** only static sensors, the source is localized during the epidemic, observations can be positive or negative.
Dynamic sensor placement	**D-OFF – Section 5.5** static and dynamic sensors, the source is localized after the epidemic, observations are always positive.	**D-ON – Section 5.6** static and dynamic sensors, the source is localized during the epidemic, observations can be positive or negative.

Table 5.1 – The source localization methods and settings considered in this chapter.

applications, ranging from localizing the source of a belief that spread in the past to tracking the source of a disease outbreak in real time, and from finding the source of a rumor in a network in which we constantly monitor a set of individuals, to detecting the patient-zero of an infection through ad-hoc interviews or clinical tests.

Experimental evaluation

Through extensive experiments on synthetic and real-world networks, we evaluate our approach along two different axes: (1) Under budget-constraints for the number of sensors, we measure the uncertainty on the identity of the source (i.e., the number of nodes that have a positive probability of being the source given the available observations); and (2) when the budget for sensors is not limited, we assess the number of sensors needed to exactly identify the source.

Our analysis highlights that a strategy that uses dynamic sensors dramatically outperforms a static strategy with the same budget: By choosing fewer than 5% of the nodes as sensors we improve the success rate of finding the source from approximately 30% to approximately 92% (see Figure 5.9a). Moreover, when we do not have a limited budget on the number of dynamic sensors, we can localize the source with a small number of sensors: between 3% and 6% of the network nodes depending on the network topology (see Figure 5.10). The reason for these improvements is that, using dynamic sensors, we

Notation

$\mathcal{S}$	set of static sensors
$\mathcal{D}$	set of dynamic sensors
$\mathcal{U}$	$\mathcal{S} \cup \mathcal{D}$, set of all sensors
K_s	budget for static sensors
K_d	budget for dynamic sensors
K	$K_s + K_d$, total budget for sensors
$\tau^\star$	time at which source localization begins
θ	deployment delay
$\omega = (u, t_u)$	observation of node u: if u is not infected, $t_u = \emptyset$
$\mathcal{B}$	set of candidate sources

can progressively reduce the network to a small sub-network whose nodes always include the source.

We also show that, given a set of constraints (how many sensors can be deployed, whether they can be deployed adaptively, ...), the choice of the sensors strongly affects the performance of source localization. In particular, we evaluate different choices of the static sensors (in Section 5.7.3) and different methods for choosing the dynamic sensors (in Section 5.7.6). We also study the effect of varying the proportion of static versus dynamic sensors, showing that we can save some resources by choosing a small budget for static sensors, but not too small as we might pay with a longer time for localizing the source (see Figure 5.12).

Finally we demonstrate that, by using all the information available as soon as it becomes available, we can greatly enhance the timeliness of source localization and restrict the search to a small set of candidate sources when the number of infected nodes is still small (see Figures 5.6a and 5.6b).

5.2 Preliminaries

5.2.1 Model

We do not assume a precise distribution $\{X_{uv}\}_{(u,v)\in E}$ for the transmission delays. Instead, we assume that there exists $\varepsilon \in [0, 1)$ such that for every $(u, v) \in E$, the support of X_{uv} is contained in $[(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}]$. We call ε the *variance parameter* because it encodes how much the transmission delays can deviate from their mean.

For $\varepsilon = 0$, $X_{uv} = w_{uv}$ for every $(u, v) \in E$, and we say that the epidemic is deterministic; for $\varepsilon > 0$, $w_{uv} \cdot \varepsilon$ gives an upper bound on the deviation of X_{uv} from its mean. By letting the maximum deviation be proportional to the edge weights we make sure that our transmission model is not trivial: If the maximum deviation was constant, the impact of the variance would depend on the scale of the edge weights and, for large w_{uv}, X_{uv} would be effectively deterministic.

In the experimental evaluation of Section 5.7, we consider mostly the case in which X_{uv} is uniformly distributed on $[w_{uv}(1-\varepsilon), w_{uv}(1+\varepsilon)]$. In this case, the variance of X_{uv} is $\mathrm{Var}(X_{uv}) = {w_{uv}^2\varepsilon^2}/{3}$ (which is the maximum variance of unimodal distributions with support $[(1-\varepsilon)w_{uv}, (1+\varepsilon)w_{uv}]$ [Seaman et al., 1985]). However we also test the performance of our methods when X_{uv} has unbounded support (see Section 5.7.9).

5.2.2 Online & Offline Source Localization

Depending on the application of interest, it can be desirable to localize the source *while* the epidemic is still spreading (e.g., for a disease spreading now) or only *after* it has propagated throughout the network (e.g., for an epidemic that happened in the past or when a timely investigation is not needed). In the former case, we speak about *online* source localization, and in the latter case about *offline* source localization. In contexts where the data becomes available while the epidemic spreads, online localization has the potential to identify the source (or a small set of candidate sources) before the epidemic propagates throughout the entire network. Offline localization, instead, is the only possible approach to source localization when we study epidemics that occurred in the past. Moreover, it is the setting that is commonly used in the literature (see Section 1.3).

Source localization might not be instantaneous. Let $\tau^\star$ be the time at which we start investigating the identity of the source and, for every $v \in V$, let t_v denote the infection time of v. We can give the following definition.

Definition 5.1 (Online/offline method)**.** *A method for source localization is said to be an online (respectively, offline) method if* $\tau^\star < \max_{v\in V} t_v$ *(resp.,* $\tau^\star \geq \max_{v\in V} t_v$*).*

The main difference between online and offline source localization is that, when performing offline localization, the full picture of the process is already available at time $\tau^\star$.

We present a framework that naturally encompasses both the offline and the online regimes. We study offline source localization in Section 5.3 and 5.5, online source localization in Section 5.4 and 5.6.

5.2.3 Sensors

Similarly to Chapters 3 and 4 we use only the information provided by a subset of nodes which we call sensors. In this chapter, sensors can be either static or dynamic.

Definition 5.2 (Static/dynamic sensor)**.** *A sensor is said to be static if it is chosen independently of any epidemic. In contrast, we say that a sensor is dynamic if it is chosen in order to localize the source of a particular epidemic.*

We assume that we have a budget K for the total number of sensors and we consider two regimes for sensor placement: static sensor placement (all K sensors are static) and dynamic sensor placement ($K_s > 0$ static sensors and $K_d = K - K_s$ dynamic sensors). Note that we never set $K_s = 0$ because otherwise no sensor would be deployed in the network when the epidemic starts spreading and the detection of the source would be trivially impossible.

The set of static (respectively, dynamic) sensors is denoted by $\mathcal{S}$ (resp., $\mathcal{D}$) and the set of all sensors is $\mathcal{U} = \mathcal{S} \cup \mathcal{D}$.

In online localization, the dynamic sensors are chosen while the epidemic spreads; in the offline regime, they are chosen while we perform source localization, i.e., by the time they are chosen the epidemic has already spread throughout the network.

As mentioned in Section 5.2.2, the localization process is generally not instantaneous because it can require a sequence of steps (e.g., updates of the estimated identity of the source when more information is available or when additional dynamic sensors are deployed). Let τ denote the time at which a step in the localization process is taken: At time τ a sensor u gives information in two possible ways: If it became infected at $t_u \leq \tau$, it reveals its infection time t_u; otherwise it informs about its susceptible state. In the first (respectively, second) case we say that the sensor gives a *positive* (resp., *negative*) *observation.*

In offline source localization, all observations are positive. Instead, in online source localization both static and dynamic sensors can give positive or negative observations and, as we will see in Section 5.4, both positive and negative observations contribute to the localization.

We represent each observation ω as a tuple $\omega \triangleq (u, t_u)$ where $u \in V$ denotes the sensor and $t_u \in \mathbb{R}$ is the infection time of u if the observation is positive, $t_u = \emptyset$ if the observation is negative.

Table 5.1 summarizes the source localization settings that we consider and the relationships between the definitions of static/dynamic sensors, online/offline source localization and positive/negative observations. Figure 5.1 illustrates our high-level approach to source localization, highlighting the different roles of static and dynamic sensors.

5.3 Offline Localization with Static Sensors (S-OFF)

In this section and in the following ones (Sections 5.4-5.6), we present the four settings listed in Table 5.1. For the sake of readability, the technical details are presented in Section 5.8.

We first describe the S-OFF algorithm with which we perform offline source localization using only static sensors. This is the setting most of the literature works with (see Section 1.3). In contrast to other approaches we are more interested in determining all nodes that are possible sources, to make sure that we did not miss the actual source, rather than in isolating only one node that would maximize the likelihood of being the source. This approach paves the way for the correctness results of D-OFF and D-ON in Sections 5.5 and 5.6.

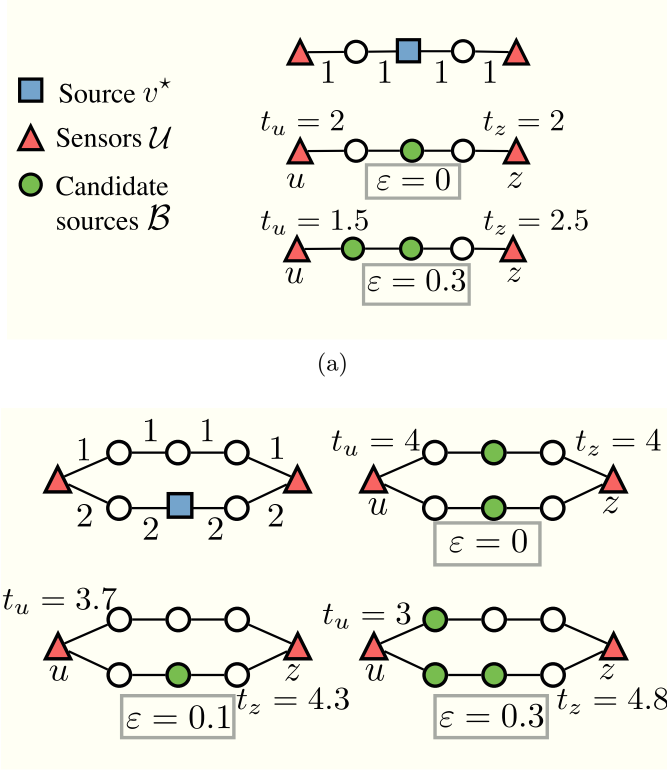
We do not use any dynamic sensor, hence $\mathcal{U} = \mathcal{S}$ and $|\mathcal{U}| = K = K_s$. As we localize the source offline, a set of positive observations of the form $\mathcal{O} = \{(u, t_u) : u \in \mathcal{U}\}$ is available at the beginning of the localization process.

Using $\mathcal{O}$, we want to determine the possible sources, i.e., the set of *candidate sources*

$$\mathcal{B} \triangleq \{v \in V : \mathbf{P}(\mathcal{O}|v^\star = v) > 0\}. \tag{5.1}$$

$\mathcal{B}$ depends not only on $v^\star$ but also on the particular realization of the infection times and on the variance parameter ε. In fact, when the variance parameter ε is larger, given a set of observations, the uncertainty on the identity of the source is higher because a larger set of nodes can initiate epidemics that result in the observed infection times. In Figure 5.2, we display the set $\mathcal{B}$ for a few simple examples. Figure 5.2a illustrates that when the infection times are not deterministic, it is in general more difficult to localize the source, as we can expect, because the infection times can substantially deviate from their mean value. More surprisingly, the reverse can also occur, as shown in the example of Figure 5.2b: For specific network topologies and moderate values of ε, non-deterministic transmission delays can make source localization easier than in the deterministic case. This is due to the maximum deviation of the transmission delay X_{uv} from w_{uv} being proportional to w_{uv} itself (see Section 5.2.1): Observing a very extreme value of the difference $t_u - t_v$ between the infection time of two sensors u and v can give information about the path along which the epidemic spread, hence about the identity of $v^\star$.

Sections 5.3.1 and 5.3.2 explain how $\mathcal{B}$ can be computed in practice.



(a)

(b)

Figure 5.2 – Examples of sets of candidate sources $\mathcal{B}$: The set $\mathcal{B}$ depends on the (unknown) source $v^\star$, on the variance parameter ε, and on the realization of the random infection times. For each of the two setups (a) and (b), the first network shows the actual source and the network topology, and the following networks show different realizations of the observations $\mathcal{O}$ for different noise parameters ε. **(a)**: For the weighted network at the top, when ε is large (bottom), $\mathcal{B}$ can be larger than for $\varepsilon = 0$ (middle). **(b)**: For the weighted network at the top-left, when ε is positive but small (bottom-left), $\mathcal{B}$ can be smaller than when $\varepsilon = 0$ (top-right); when ε is large (bottom-right) $\mathcal{B}$ can be larger than in the two previous cases.

5.3.1 Deterministic Epidemics

We explained in Section 1.2.2 that, when the starting time $t^\star$ of the epidemic is unknown, no single observation taken in isolation is informative about the identity of the source. Instead, a set of two (or more) observations gives information on the identity of the source. Therefore we start defining, for two observations ω_1, ω_2, the event of observing ω_1 and ω_2 jointly.

Definition 5.3 (Event A_{ω_1,ω_2}). *Let* $\omega_1 \triangleq (u_1, t_{u_1})$*, and* $\omega_2 \triangleq (u_2, t_{u_2})$*,* $\omega_1 \neq \omega_2$*, be two observations. We define the event* A_{ω_1,ω_2} *as* $A_{\omega_1,\omega_2} \triangleq \{T(v^\star, u_1) - T(v^\star, u_2) = t_{u_1} - t_{u_2}\}$.

For every pair of observations ω_1, ω_2, $\omega_1 \neq \omega_2$, we define

$$\mathcal{B}_{\omega_1,\omega_2} \triangleq \{v \in V : \mathbf{P}(A_{\omega_1,\omega_2} | v^\star = v) > 0\}. \tag{5.2}$$

When epidemics are deterministic ($\varepsilon = 0$), $\mathcal{B}$ can be easily computed using the result of the following proposition (see Section 5.8.1 for proof and complementary lemmas).

Proposition 5.4. *Let* $\mathcal{O}$ *be a set of observations and let* $\varepsilon = 0$*. Let* $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}$ *be a fixed observation, which we call the reference observation. Then, the set of candidate sources* $\mathcal{B}$ *is*

$$\mathcal{B} = \bigcap_{\omega \in \mathcal{O} \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}. \tag{5.3}$$

From Proposition 5.4, we can compute the candidate set $\mathcal{B}$ with Algorithm 6.

Algorithm 6 S-OFF - deterministic epidemic

```
Require: $\mathcal{O}$ set of observations
  $\mathcal{B} \leftarrow V$
  $\omega_1 \triangleq (u_1, t_{u_1}) \leftarrow$ Sample($\mathcal{O}$)
  for $(u, t_u) \in \mathcal{O} \setminus \{\omega_1\}$ do
      for $v \in \mathcal{B}$ do
          if $d(v, u) - d(v, u_1) \neq t_u - t_{u_1}$ then
              remove $v$ from $\mathcal{B}$
  return $\mathcal{B}$
```

The runtime of Algorithm 6 is $O(K_s N)$. When $v^\star = v$, since $T(v, u_i) - T(v, u_1) = t_i - t_1$ is deterministic and equal to $d(v, u_i) - d(v, u_1)$ for any $u_1, u_i \in \mathcal{U}$, the set $\mathcal{B}$ of candidate sources returned by Algorithm 6 is equal, because of Lemma 1.9, to $[v]_{\mathcal{U}}$. Hence, in the inner **for** of Algorithm 6 it would be enough to loop over a set of representatives v of the equivalence classes in $\mathcal{B}$. However, for consistency with the algorithms presented in the following sections, we keep the version of the algorithm given in Algorithm 6, where the loop is over all $v \in \mathcal{B}$.

5.3.2 Non-deterministic Epidemics

When the transmission delays are not deterministic ($0 < \varepsilon < 1$), verifying analytically if $\mathbf{P}(\mathcal{O}|v^\star = v) > 0$ is computationally intense for two reasons: the interdependence of the events $\{A_{\omega_i,\omega_j}\}_{\omega_i \neq \omega_j}$ and, in meshed networks, the multiplicity of possible propagation paths. To overcome these difficulties, we compute a superset $\widetilde{\mathcal{B}} \supseteq \mathcal{B}$ of the set of candidate sources using the result of the following proposition.

Proposition 5.5. *Let* $0 < \varepsilon < 1$*, let* $\omega_1 \triangleq (u_1, t_{u_1})$*,* $\omega_2 \triangleq (u_2, t_{u_2}) \in \mathcal{O}$*,* $\omega_1 \neq \omega_2$*, and let* $v \in \mathcal{B}$*. Then*

$$|d(v, u_1) - d(v, u_2) - t_{u_1} + t_{u_2}| \leq \varepsilon(d(v, u_1) + d(v, u_2)). \tag{5.4}$$

Based on Proposition 5.5, we define

$$\widetilde{\mathcal{B}} \triangleq \Big\{ v \in V : |d(v, u_1) - d(v, u_2) - t_{u_1} + t_{u_2}| \leq \varepsilon(d(v, u_1) + d(v, u_2)) \\ \forall (u_1, t_{u_1}), (u_2, t_{u_2}) \in \mathcal{O} \Big\}. \tag{5.5}$$

Remark 5.6. *The runtime of computing* $\widetilde{\mathcal{B}}$ *is* $O(K_s^2 N)$*. In fact, in contrast to Algorithm 6, we need to loop over all pairs* $(\omega_1, \omega_2) \in \mathcal{O} \times \mathcal{O}$ *with* $\omega_1 \neq \omega_2$*. If we fix a reference observation* ω_1 *as in Proposition 5.4 and, for a candidate source* v*, we verify* (5.4) *only for the pairs* $(\omega_1, \omega_i), \omega_i \in \mathcal{O}$*, we would obtain a larger set of candidate sources that might also includes some nodes* v *such that* $\mathbf{P}(\mathcal{O}|v^\star = v) = 0$*.*

In fact, given $v \in V$ *and* $\omega_1 \triangleq (u_1, t_1), \omega_i \triangleq (u_i, t_i), \omega_j \triangleq (u_j, t_j) \in \mathcal{O} \backslash \{\omega_1\}$*,* $\omega_i \neq \omega_j$*, it is possible that*

$$\begin{cases} |d(u_1, v) - d(u_i, v) - t_1 + t_i| \leq \varepsilon(d(u_1, v) + d(u_i, v)) \\ |d(u_1, v) - d(u_j, v) - t_1 + t_j| \leq \varepsilon(d(u_1, v) + d(u_j, v)) \end{cases} \tag{5.6}$$

and yet $|d(u_i, v) - d(u_j, v) - t_i + t_j| > \varepsilon(d(u_1, v) + d(u_j, v))$*.*

We now give a concrete example. Consider the network (with sensor infection times) of Figure 5.3. If $\varepsilon = 0.25$*, for node* v *we have*

$$|d(v, u) - d(v, z) - t_u + t_z| \leq \varepsilon(d(v, u) + d(v, z)) \tag{5.7}$$

and

$$|d(v, u) - d(v, w) - t_u + t_w| \leq \varepsilon(d(v, u) + d(v, w)) \tag{5.8}$$

but

$$|d(v, w) - d(v, z) - t_w + t_z| > \varepsilon(d(v, w) + d(v, z)). \tag{5.9}$$

Hence, taking u as reference sensor and comparing t_u with t_z and t_w we would not remove v from the set of candidate sources, which instead we do if we further compare t_w and t_z.

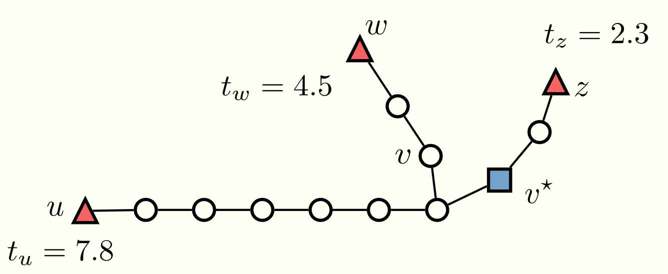


Figure 5.3 – When $\varepsilon > 0$, taking a single reference sensor, we get a larger set of candidate sources. In this example, taking u as reference sensor and comparing t_u with t_z and t_w we would not remove v from the set of candidate sources, which instead we do if we further compare t_w and t_z.

Remark 5.7. *The set $\widetilde{\mathcal{B}}$ defined in* (5.5) *is, in general, strictly larger than $\mathcal{B}$. When computing $\widetilde{\mathcal{B}}$ we consider only two observations at a time; however, if ω_1, ω_2 and ω_3 are three distinct observations, it is possible that $\mathbf{P}(v^\star = v|\omega_i, \omega_j) > 0$ for every $i, j \in \{1, 2, 3\}$ but $\mathbf{P}(v^\star = v|\omega_1, \omega_2, \omega_3) = 0$ (similar situations can arise for larger sets of observations). When epidemics are non-deterministic, verifying if $\mathbf{P}(v^\star = v|\mathcal{O}) > 0$ for a set $\mathcal{O}$ of arbitrary cardinality would be computationally intractable, roughly exponential in the cardinality of the set. Thus, we approximate $\mathcal{B}$ with $\widetilde{\mathcal{B}}$. Section 5.7 demonstrates empirically that the size of $\widetilde{\mathcal{B}}$ decreases very fast over the iterations of the algorithm, indicating that our approximation is not too loose.*

We conclude this section with a proposition stating that, for low ε and $v^\star = v$, $\widetilde{\mathcal{B}} \subseteq [v]_{\mathcal{U}}$. This guarantees that when ε is sufficiently small, identifying the source is at least as easy as in the deterministic case.

Proposition 5.8. *Let $\mathcal{U}$ be the sensor set. Let*

$$\Delta \triangleq \max_{u \in \mathcal{U}, v \in V} d(v, u)$$

and

$$\phi \triangleq \min_{[v_1]_{\mathcal{U}} \neq [v_2]_{\mathcal{U}}} \max_{u_1, u_2 \in \mathcal{U}} |d(v_1, u_1) - d(v_1, u_2) - d(v_2, u_1) + d(v_2, u_2)|.$$

If $\varepsilon < \varepsilon_0 \triangleq \phi/4\Delta$ and $v^\star = v$, then $\widetilde{\mathcal{B}} \subseteq [v]_{\mathcal{U}}$.

If additional conditions on the edge weights or on the network topology are given, more refined bounds ε_0 in Proposition 5.8 can be derived. For example, in a *tree* with weights

$w_{uv} > C \in \mathbb{R}^+$, the uniqueness of the path between two any nodes yields that $\phi \geq 2C$ for every $\mathcal{U}$. Hence, in this case, the statement holds for $\varepsilon < C/2\Delta$ (see also Proposition 3.1).

5.4 Online Localization with Static Sensors (S-ON)

In online localization with static sensors (S-ON), the localization of the source can be seen as a process in which we iteratively refine the set $\mathcal{B}$ of candidate sources while we gather more and more information about the epidemic.

Given a static sensor set $\mathcal{U}$, the final outcome of S-ON and S-OFF is identical in terms of the nodes that are identified as candidate sources. The difference is the ability of S-ON to restrict the search for the source to a very small subset of nodes when the epidemic has not yet propagated throughout the network.

In contrast to Section 5.3, where the observation set $\mathcal{O}$ contained all the observations available *at the end* of the epidemic, which could therefore only be positive, here $\mathcal{O}$ changes *while* the epidemic progresses. From a technical point of view, the difference between S-ON and S-OFF is that in S-ON some observations are negative. Hence the results of Proposition 5.4 (respectively, 5.5) must be refined to include in the computation of $\mathcal{B}$ (resp., $\widetilde{\mathcal{B}}$) the information given by negative observations. We start the localization process as soon as a sensor gets infected, i.e., at time $\tau^\star \triangleq \min_{u \in \mathcal{U}} t_u$. We denote by $\mathcal{O}_t$ the observation set at time t. At every time $t \geq \tau^\star$, $\mathcal{O}_t$ is the union of a set of positive observations and of a set of negative observations (see Section 5.2.3). We denote by $\mathcal{O}_t^+$ (respectively, $\mathcal{O}_t^-$) the set of positive (respectively, negative) observations at time t. This means that, for every $t \geq \tau^\star$ we have

$$\mathcal{O}_t^+ = \{(u, t_u) : u \in \mathcal{U}, t_u \leq t\},$$

$$\mathcal{O}_t^- = \{(u, \emptyset) : u \in \mathcal{U}, t_u > t\},$$

and $\mathcal{O}_t = \mathcal{O}_t^+ \cup \mathcal{O}_t^-$.

Definition 5.9 (Event $A^t_{\omega_1,\omega_2}$)**.** *Let* $t \in \mathbb{R}$ *and let* $\omega_1 \triangleq (u, t_u) \in \mathcal{O}_t^+$, $\omega_2 \triangleq (w, \emptyset) \in \mathcal{O}_t^-$ *be two observations, one positive and one negative. We define the event* $A^t_{\omega_1,\omega_2}$ *as* $A^t_{\omega_1,\omega_2} \triangleq \{T(v^\star, u) - T(v^\star, w) < t_u - t\}$.

Note that, if $\omega_1 \neq \omega_2$ are two negative observations in $\mathcal{O}_t^-$, for every possible source there exists a starting time $t^\star \in \mathbb{R}$ such that both ω_1 and ω_2 hold at time t (and this remark generalizes to larger sets of negative observations). For this reason, sets of only negative observations are not useful to localize the source.

If only negative observations are available, we do not even know if there is an ongoing epidemic. However, combining negative observations and positive observations we can design an algorithm, S-ON, which, at any time t during the localization process, computes the smallest possible set of candidate sources: Given the information contained in $\mathcal{O}_t$, the set of candidates $\mathcal{B}_t$ computed by S-ON contains all and only the nodes that have a positive probability of being the source.

More formally, we define the candidate sources set at time t as

$$\mathcal{B}_t \triangleq \{v \in V : \mathbf{P}(\mathcal{O}_t | v^\star = v) > 0\}. \tag{5.10}$$

For every pair of positive observations ω_1, ω_2, $\omega_1 \neq \omega_2$, let $\mathcal{B}_{\omega_1,\omega_2}$ be defined as in (5.2). At time t, for every positive observation ω_1 and negative observation ω_2 we also define

$$\mathcal{B}_{\omega_1,\omega_2,t} \triangleq \{v \in V : \mathbf{P}(A^t_{\omega_1,\omega_2} | v^\star = v) > 0\}.$$

Then Proposition 5.4 can be extended as follows (see Section 5.8.2 for proof and complementary lemmas).

Proposition 5.10. *Let $t \in \mathbb{R}$, $\mathcal{O}_t$ be the set of observations at time t and $\varepsilon = 0$. Let $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}^+_{\tau^\star}$ be the first positive observation that we call the reference observation. Then, the set of candidate sources $\mathcal{B}_t$ is*

$$\mathcal{B}_t = \left(\bigcap_{\omega \in \mathcal{O}^+_t \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega} \right) \bigcap \left(\bigcap_{\omega \in \mathcal{O}^-_t} \mathcal{B}_{\omega_1,\omega,t} \right).$$

Moreover, if $t, t' \in \mathbb{R}$, $t' > t$, $\mathcal{B}_{t'} \subseteq \mathcal{B}_t$.

Call $\tau^\star = t_1 < t_2 < .. < t_F = \tau_F$ the times at which the observation set changes. Denoting by t_u the infection time of sensor u we have $\tau_F \triangleq \max_{u \in \mathcal{U}} t_u$. Let us also denote $\mathcal{O}_i \triangleq \mathcal{O}_{t_i}$.

Using the result of Proposition 5.10 we compute and update the set of candidate sources with Algorithm S-ON (see Algorithm 7). S-ON updates the set of candidate sources $\mathcal{B}$ at every time t_i, $1 \leq i \leq F$, producing a set that we call $\mathcal{B}_i$.

At every time step t_i, S-ON produces the smallest possible set of candidate sources that always contain the source $v^\star$. More formally, we have the following.

Proposition 5.11. *For every $1 \leq i \leq F$ there is no algorithm $\mathcal{A}$ different from S-ON which, given $\mathcal{O}_i$, produces a set of candidate sources $\mathcal{B}_i(\mathcal{A}) \subsetneq \mathcal{B}_i$ and $\mathbf{P}(v^\star \in \mathcal{B}_i(\mathcal{A})) = 1$.*

Proof. By Proposition 5.10, the set $\mathcal{B}_i$ produced by S-ON is equal to $\mathcal{B}_{t_i}$ for all $1 \leq i \leq F$. Hence, by (5.10), for every $v \in \mathcal{B}_i$, $\mathbf{P}(v^\star = v | \mathcal{O}_i) > 0$. If an algorithm $\mathcal{A}$ produces $\mathcal{B}_i(\mathcal{A}) \subsetneq \mathcal{B}_i$ there exists $v \in \mathcal{B}_i \setminus \mathcal{B}_i(\mathcal{A})$ such that $\mathbf{P}(v^\star = v | \mathcal{O}_i) > 0$. Therefore $\mathbf{P}(v^\star \in \mathcal{B}_i(\mathcal{A})) < 1$. □

Like for Algorithm 6, Algorithm 7 runs in time $O(K_s N)$.

Algorithm 7 S-ON - deterministic epidemic

```
Require: Observation sets $\{\mathcal{O}_i^+\}_{i=1}^F$, $\{\mathcal{O}_i^-\}_{i=1}^F$
  $\mathcal{B}_0 \leftarrow V$
  $\omega_1 \triangleq (u_1, t_{u_1}) \leftarrow$ Sample($\mathcal{O}_1^+$)
  $\mathcal{O}_0^+ \leftarrow \{\omega_1\}$
  $i \leftarrow 1$
  while $i \leq F$ and $|\mathcal{B}_{i-1}| > 1$ do
      $i \leftarrow i + 1$
      $\mathcal{B}_i \leftarrow \mathcal{B}_{i-1}$
      for $(u, t_u) \in \mathcal{O}_i^+ \backslash \mathcal{O}_{i-1}^+$ do
          for $v \in \mathcal{B}_i$ do
              if $d(u, v) - d(u_1, v) \neq t_u - t_{u_1}$ then
                  remove $v$ from $\mathcal{B}_i$
      for $(u, \emptyset) \in \mathcal{O}_i^-$ do
          for $v \in \mathcal{B}_i$ do
              if $d(u, v) - d(u_1, v) < t_i - t_{u_1}$ then
                  remove $v$ from $\mathcal{B}_i$
  return $\mathcal{B}_i$
```

The extension of S-ON to $\varepsilon > 0$ follows similarly to Section 5.3.2 and is presented in Section 5.8.2.

5.5 Offline Localization with Static and Dynamic Sensors (D-OFF)

We now study D-OFF: offline source localization with static and dynamic sensors. After computing the set of candidate sources $\mathcal{B}$ by using the observations gathered by the static sensors, we use dynamic sensors to refine $\mathcal{B}$, i.e., to remove as many nodes as possible from $\mathcal{B}$. Hence our D-OFF algorithm is, in its first part, identical to Algorithm 6 for S-OFF whereas, in its second part, it consists of an iterative refinement of $\mathcal{B}$ based on the observations obtained through the newly-deployed dynamic sensors. If we obtain a candidate sources set $\mathcal{B}$ such that $|\mathcal{B}| = 1$ before deploying the entire budget K_d, we stop deploying dynamic sensors.

Clearly, not all possible dynamic sensors are equally informative about the identity of the source. Our strategy is to iteratively choose where to place the dynamic sensors in order to maximize the progress in the localization of the source, which we refer to as GAIN. In Section 5.5.2 we compare three possible notions of GAIN.

In Algorithm 8 we present the pseudo-code for deterministic epidemics. The extension to non-deterministic epidemics follows directly from the results of Section 5.3.2.

Algorithm 8 D-OFF - deterministic epidemic

```
Require: $\mathcal{O}$ set of observations, $K_d$ budget for dynamic sensors
  $\mathcal{B} \leftarrow V$
  $\omega_1 \triangleq (u_1, t_1) \leftarrow$ Sample($\mathcal{O}$)
  for $(u, t) \in \mathcal{O} \backslash \omega_1$ do
      for $v \in \mathcal{B}$ do
          if $d(u, v) - d(u_1, v) \neq t - t_1$ then
              remove $v$ from $\mathcal{B}$
  $\mathcal{B}_0 \leftarrow \mathcal{B}$
  $i \leftarrow 0$
  while $|\mathcal{B}_i| > 1$ and $i < K_d$ do
      $i \leftarrow i + 1$
      $d_i \leftarrow \text{argmax}_{d \in V \backslash \mathcal{U}} \textsc{Gain}_{\mathcal{U}}(d)$
      $\mathcal{U} \leftarrow \mathcal{U} \cup \{d_i\}$
      $t_{d_i} \leftarrow$ infection time of $d_i$
      $\mathcal{B}_i \leftarrow \mathcal{B}_{i-1}$
      for $v \in \mathcal{B}_i$ do
          if $d(d_i, v) - d(u_1, v) \neq t_{d_i} - t_1$ then
              remove $v$ from $\mathcal{B}_i$
  return $\mathcal{B}_i$
```

Let $\chi_{\textsc{Gain}}$ denote the time required to compute Gain. The runtime of Algorithm 8 is $O(N(K_s + \chi_{\textsc{Gain}} K_d))$.

5.5.1 Correctness

If we could observe the infection time of all nodes in V, we could identify the source trivially by looking at the node with the smallest infection time. We now prove that when the budget for dynamic sensors is unrestricted, Algorithm 8 converges to the set containing only the source $v^\star$ independently of the variance parameter ε. In other words, Algorithm 8 never misses the source.

This result is tight in the budget $K_s + K_d$ of dynamic sensors (as stated in Section 1.2.2, for some network topologies the number of sensors needed to localize the source can go up to $N - 1$) Proving tighter results for particular classes of network topologies is an interesting direction for future work.

The correctness of Algorithm 8 does not depend on the definition of Gain: As we will see in Section 5.7, Gain has an effect on the number of sensors required in Algorithm 8 but not on its correctness.

Theorem 5.12. *Let $\varepsilon \in [0, 1)$ and X_{uv} be a 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 ($K_s + K_d = N$). Algorithm 8 always returns $\{v^\star\}$.*

Proof. We prove the statement for $\varepsilon > 0$, the proof for $\varepsilon = 0$ can be derived in a similar way. First, note that nodes are removed from the set of candidate sources if and only if they do not satisfy, for some u_1, u_2, the necessary conditions expressed by (5.4). Hence, due to Proposition 5.5, the source $v^\star$ is never removed from the set of candidates. Next, we want to prove that, for every node $w \neq v^\star$, there exist $z, y \in V$ such that, when the infection times of z, y are observed, w is removed from the set of candidate sources. Suppose that $v^\star = v$ and that its infection time t_v is observed. Let $w \neq v$ be another node for which the infection time t_w is also observed. As $v^\star = v$, we have $t_w > t_v$. Note that (5.4) cannot hold for w with $u_1 = w$ and $u_2 = v$: Indeed, we would have $0 < t_w - t_v \leq (\varepsilon - 1)d(v, w) < 0$, which gives a contradiction. Let $i' \in \mathbb{N}^+$ such that, when $i = i'$ in Algorithm 8, both $v \in \mathcal{U}$ and $w \in \mathcal{U}$. Then, $w \notin \mathcal{B}_{i'}$. $\square$

5.5.2 Natural Gain Functions

We consider three possible Gain functions to be used for the selection of the dynamic sensors.

Size-Gain. Perhaps the most natural Gain function is the one that computes the expected reduction in the number of candidate sources. Let $\mathcal{B}_\mathcal{U}$ denote the set of candidate sources computed based on the information given by the sensors in $\mathcal{U}$ and $\mathcal{B}^c_\mathcal{U} \subseteq \mathcal{B}_\mathcal{U}$ the set of candidate sources after adding $c \in V \backslash \mathcal{U}$ as a dynamic sensor. We define the Size-Gain of choosing c as a dynamic sensor as $g^{\text{SIZE}}_\mathcal{U}(c) \triangleq \mathbf{E}[|\mathcal{B}_\mathcal{U}| - |\mathcal{B}^c_\mathcal{U}|]$. Hence, maximizing $g^{\text{SIZE}}_\mathcal{U}$ is equivalent to minimizing the size of $\mathcal{B}^{(c)}_\mathcal{U}$ and maximizing $g^{\text{SIZE}}_\mathcal{U}$ gives, at any step, a sensor choice that is locally optimal.

For deterministic epidemics, $g^{\text{SIZE}}_\mathcal{U}(c)$ can be easily computed by summing over the set $\mathcal{T}^c_\mathcal{U}$ of the possible infection times for c (see Definition 5.13 below). For $\varepsilon \in (0, 1)$ we propose an approximation of $g^{\text{SIZE}}_\mathcal{U}(c)$ in Section 5.8.5.

Definition 5.13 (Possible infection times). *Let $\mathcal{U}$ be a set of sensors, $c \in V \backslash \mathcal{U}$, $\mathcal{O}_\mathcal{U} \triangleq \{(u, t_u), u \in \mathcal{U}\}$ and fix $(u_1, t_1) \in \mathcal{O}_\mathcal{U}$ arbitrarily. Let $\mathcal{B}_\mathcal{U}$ be the set of candidate sources after observing the infection times of the nodes in $\mathcal{U}$, i.e., $\mathcal{B}_\mathcal{U} = \{v \in V : \mathbf{P}(v = v^\star | \mathcal{O}_\mathcal{U}) > 0\}$. Then*

$$\mathcal{T}^c_\mathcal{U} \triangleq \{h \in \mathbb{R} : h = d(v, c) - d(v, u_1) - t_1 \text{ for some } v \in \mathcal{B}_\mathcal{U}\}$$

is the set of possible infection times of c.

Note that when $\varepsilon = 0$, the cardinality of $\mathcal{T}^c_\mathcal{U}$ is always finite and equal to the number of equivalence classes in which $\mathcal{U} \cup \{c\}$ partitions $\mathcal{U}$ (see Definition 1.3). With techniques similar to those of the proof of Proposition 5.4, it is easy to prove that Definition 5.13 does not depend on the choice of $(u_1, t_1) \in \mathcal{O}_\mathcal{U}$. The next proposition shows how $g^{\text{SIZE}}_\mathcal{U}$ can be computed in practice.

Proposition 5.14. *Let* $\mathcal{U}$ *be a set of sensors,* $c \in V \backslash \mathcal{U}$*,* $\mathcal{O}_{\mathcal{U}}, \mathcal{B}_{\mathcal{U}}$ *as in Definition 5.13 and fix* $(u_1, t_1) \in \mathcal{O}_{\mathcal{U}}$ *arbitrarily. Call* t_c *the infection time of* c *and define*

$$\begin{aligned} b_{\mathcal{U}}(c,h) &\triangleq \{v \in \mathcal{B}_{\mathcal{U}} : \mathbf{P}(v = v^\star | t_c = h) > 0\} \\ &= \{v \in \mathcal{B}_{\mathcal{U}} : h = d(v,c) - d(v,u_1) + t_1\}. \end{aligned}$$

Then,

$$g_{\mathcal{U}}^{\text{SIZE}}(c) = \sum_{h \in \mathcal{T}_c} \mathbf{P}(v^\star \in b_{\mathcal{U}}(c,h)) \cdot (|\mathcal{B}_{\mathcal{U}}| - |b_{\mathcal{U}}(c,h)|). \tag{5.11}$$

Proof. Follows from the definition of $g_{\mathcal{U}}^{\text{SIZE}}$, $\mathcal{T}_c$ and $b_{\mathcal{U}}$. □

DRS-Gain. The definition of this second Gain function is inspired by the notion of double resolving ser (DRS, see Section 1.2.2). When epidemics spread deterministically, observing the infection times of a DRS of the candidate-sources set $\mathcal{B}$ removes all ambiguities about the source identity. With DRS-Gain, we iteratively choose the sensor that, added to the current sensor set $\mathcal{U}$, gives the most progress in forming a DRS of $\mathcal{B}$.

Let $c \in V \backslash \mathcal{U}$ and $\mathcal{T}_{\mathcal{U}}^c$ as in Definition 5.13. Then, the DRS-Gain of adding c to $\mathcal{U}$ is

$$g_{\mathcal{U}}^{\text{DRS}}(c) \triangleq |\mathcal{T}_{\mathcal{U}}^c|. \tag{5.12}$$

Since there is no direct extension of $g_{\mathcal{U}}^{\text{DRS}}$ to the non-deterministic case, we use the above definition of $g_{\mathcal{U}}^{\text{DRS}}$ independently of the variance parameter ε.

RC-Gain. RC-Gain (random candidate) assigns gain 1 to all candidates sources and gain 0 to all nodes that are not candidate sources, i.e., when the sensor set is $\mathcal{U}$ and $\mathcal{B}_{\mathcal{U}}$ is the set of candidate sources, for $c \in V \backslash \mathcal{U}$ we set $g_{\mathcal{U}}^{\text{RC}}(c) = 1$ if $c \in \mathcal{B}_{\mathcal{U}}$, $g_{\mathcal{U}}^{\text{RC}}(c) = 0$ 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 $\mathcal{B}_{\mathcal{U}}$ is already observed, adding a sensor in any other node in $\mathcal{B}_{\mathcal{U}}$ implies $|\mathcal{B}_{\mathcal{U} \cup \{c\}}| < |\mathcal{B}_{\mathcal{U}}|$, independently of the variance parameter ε.[1] Hence, this very simple Gain ensures that the localization of the source makes progress whenever a new dynamic sensor is chosen.

For any of the proposed Gain functions, the computation time χ_{Gain} is $O(|\mathcal{B}|) \subseteq O(N)$.

As it is not *a priori* clear which version of Gain leads to a better performance of Algorithm 8, in Section 5.7 we experiment with all of them.

---

[1] This can be proven with an argument analogous to the one used in the proof of Theorem 5.12: If the infection time of two nodes is observed, only one of the two (the one with smaller infection time) can belong to the set of candidate sources $\mathcal{B}$ (or, for $\varepsilon \in (0,1)$, to the superset $\widetilde{\mathcal{B}}$).

5.6 Online Localization with Static and Dynamic Sensors (D-ON)

We now turn to the online version of D-OFF: D-ON.

As in Section 5.4, we set the time $\tau^\star$ at which the localization starts to the earliest time at which a static sensor gets infected, i.e., $\tau^\star = \min_{s \in \mathcal{S}} t_s$. Starting from $\tau^\star$, we run online source localization as per S-ON and, in addition, we deploy dynamic sensors one after the other to refine the localization of the source till the budget K_d for dynamic sensors is exhausted. Specifically, a new dynamic sensor is deployed at times $\tau^\star + j\theta$, $j \in \{1, .., K_d\}$, where $\theta \in \mathbb{R}^+$ is fixed. We call θ the *deployment delay.* The choice of θ, which will be discussed in Section 5.7, requires the evaluation of the trade-off between timely localization and resource savings: With a large θ we are likely to have less negative observations, hence to reach the localization with few dynamic sensors, but a long time after the beginning of the epidemic. Vice versa, with a small θ, we are likely to reach the localization earlier but by deploying more dynamic sensors.

At time t, the candidate set $\mathcal{B}$, the sensor set $\mathcal{U}$, and the observation set $\mathcal{O} = \mathcal{O}^+ \cup \mathcal{O}^-$ are updated in two cases:

I) if $t = \tau^\star + j\theta$, $j \in \mathbb{N}$, i.e., at time t a new dynamic sensor is added;

II) if $t = t_u > \tau^\star$, i.e., t is the infection time of a static sensor or of a node that was chosen as dynamic sensor before time t but was not infected before time t.

For both cases, technical details are given in Section 5.8.3. Note that D-ON includes D-OFF as a special case. If we run D-ON starting at $\tau^\star > \max_{v \in V} t_v$, the initial observations set is $\mathcal{O}_1 = \{(s, t_s) : s \in \mathcal{S}\}$. Moreover, throughout the process all observations are positive and we recover D-OFF.

5.6.1 Correctness

The correctness result of Theorem 5.12 holds also for D-ON.

Theorem 5.15. *Let $\varepsilon \in [0, 1)$ and X_{uv} be a 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 ($K_s + K_d = N$). D-ON always returns $\{v^\star\}$.*

Proof. The proof is almost identical to that of Theorem 5.12, the only necessary change is in the last step. With the notations of the proof of Theorem 5.12, at the minimum time t such that $(v, t_v), (w, t_w) \in \mathcal{O}_t^+$, it is guaranteed that $w \notin \mathcal{B}$. □

Finally, online localization needs, on average, more resources to reach convergence to the source with respect to offline localization. This is due to the fact that, when running offline localization, every deployed sensor can directly reveal its infection time (i.e., there are no negative observations). As in the choice of the deployment delay θ, in order to choose between D-ON and D-OFF, the trade-off between resource savings and timeliness must be evaluated.

For the extension of the gain functions of Section 5.5.2 to the online setting (in which we can have negative observations) we refer the reader to Section 5.8.4.

5.7 Experimental Results

5.7.1 Experimental Setup

Transmission delays

In our experiments, the *transmission delays* are *uniformly distributed* (except in Section 5.7.9). The uniform distribution is, among the unimodal distributions on a bounded support, the one that maximizes the variance [Gray and Odell, 1967], which makes it a very challenging setting for source localization.

Static sensors

We choose the K_s static sensors with one of the following two rules:

- ◇ KDRS (K-nodes approximation of a double resolving set): This rule computes the set of K_s sensors that maximize the number of equivalence classes (see Section 1.2.2). In Chapter 4, where this sensor placement was presented under the name of LV-Obs, we showed that KDRS outperforms several common heuristics for sensor placement in the case of deterministic or low-variance epidemics.
- ◇ KMED (K-Medians): This rule computes the optimal placement of K_s sensors for the closely-related problem of maximizing the detectability of a flow [Berry et al., 2006]. The KMED placement is the set of K_s nodes $\mathcal{S}$ such that

  $$\mathcal{S} = \text{argmin}_{|\mathcal{S}|=K_s} \sum_{v \in V} \min_{s \in \mathcal{S}} d(v, s).$$

  Determining the K-Medians of a network is NP-hard [Kariv and Hakimi, 1979], hence we approximate KMED with a greedy heuristic. Contrary to KDRS sensors, which are generally placed in the periphery of the network, KMED sensors are more uniformly spread.

	ER (p=0.016)	BA (m=2)	RGG (R=0.3)	RT	PLT
$\lvert V\rvert$	250	250	250	250	250
$\lvert E\rvert$	511	496	696	249	249
avg degree	4.09	3.96	5.6	1.99	1.99
avg shortest path	4.09	3.47	9.68	7.45	37.8
avg clustering	0.02	0.06	0.56	0	0

	FB	U-WAN	WAN
$\lvert V\rvert$	3732	2258	2258
$\lvert E\rvert$	82305	17695	17695
avg degree	44.1	15.67	15.67
avg shortest path	5.34	6.94	3.56
avg clustering	0.54	0.65	0.65

Table 5.2 – Statistics for the networks used in the experiments.

In Chapter 4 we showed that the optimal placement of static sensors depends on the variance parameter ε and that, given a value of ε, a suitable sensor placement can be found interpolating between KDRS and KMED. For this reason, we limit ourselves to considering these two alternative static-sensor placements, and we evaluate their respective benefits for the different localization settings.

Default parameters

If it is not differently specified, we set the budget for static sensors to $K/N = 2\%$, the gain function for the choice of the dynamic sensors to SizeGain and the deployment delay to $\theta = 0.5$. The reasons for these choices will be clear in the following discussion.

All results are averaged over at least 100 simulations in which the source is chosen uniformly at random.

For readability, throughout this section the set of candidate sources is always denoted by $\mathcal{B}$ even when $\varepsilon > 0$ and we are actually computing the approximate set $\widetilde{\mathcal{B}}$.

5.7.2 Network Topologies

We experiment with both synthetic and real-world networks; the network properties and statistics are reported in Table 5.2.

Synthetic networks

We generated synthetic networks from the following classes: Erdős-Rényi networks (ER) [Erdős and Rényi, 1959], Barabási-Albert networks (BA) [Barabási and Albert, 1999], random geometric networks on the sphere (RGG) [Penrose, 2003], 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 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 [McAuley and Leskovec, 2012], after removing the ego nodes[2] and taking the largest connected component.

World Airline Network (WAN). This network is obtained from a publicly available dataset [OpenFlights] that provides the aircraft type used for every daily connection between over three thousands airports. Using these data we can derive the number of seats available on each route daily. We preprocess the network by removing the connections on which fewer than 20 seats per day are available and by assigning to each connection uv the average between the number of seats available from u to v and from v to u. Also, we iteratively remove the leaves (for which we believe connections are not well represented in the dataset), and we obtain a network of 2258 nodes.

Edge weights

All our results are valid for arbitrary edge weights $w_{uv} \in \mathbb{R}^+$. For our experiments we consider integer edge weights for several reasons. When $\varepsilon = 0$, integer weights actually make the problem more challenging because it is more difficult to distinguish among nodes based on their distances to the sensors; when $\varepsilon > 0$ instead, taking integer weights does not affect the difficulty of the problem because network distances cannot anyway be precisely recovered from the observations. Moreover, in practice, distances are always known up to some degree of precision and, up to a multiplication factor, edge weights can always be assumed to be integer.

All synthetic networks and the FB network are given unit edge weights as there is not a straightforward method for deriving realistic edge weights for these networks. For WAN the definition of the edge weights is inspired by a work by Colizza et al. [2006]. An edge uv is weighted with an approximation of the expected time between the infection of city u and the arrival of an infected individual at city v (see below for details). This gives a

---

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

very skewed weight distribution. Our experiments show that the variability 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).

Weights for the WAN Network. Our definition of the edge weights for the WAN network is inspired by the work of Colizza et al. [2006].

Let s_{ij} be the number of seats available on a flight from airport i to airport j. The number of seats can be inferred by the aircraft with which the flight is operated [OpenFlights]. Moreover, let $\alpha = 0.7$ denote the average occupancy rate on a flight [Colizza et al., 2006] and N_i denote the population of city i. We approximate the probability that an individual flies from i to j as $\alpha s_{ij}/N_i$.

Let θ be the probability that an individual is infected when the infection reached the city where he leaves. Then the probability that a sick individual travels from i to j is $1 - (1 - \alpha s_{ij}/N_i)^{\theta N_i}$. Hence the average delay for the infection to spread from city i to city j can be estimated to be

$$w_{ij} = [1 - (1 - \alpha s_{ij}/N_i)^{\theta N_i}]^{-1} \approx [1 - \exp^{-\alpha s_{ij}\theta}]^{-1}. \tag{5.13}$$

We assume $\theta = 0.05$ and, as discussed above, we rounded all weights w_{ij} to the closest integer. Figure 5.4 shows the resulting weight distribution (note the log-scale of the y-axis, hence the skewness of the distribution).

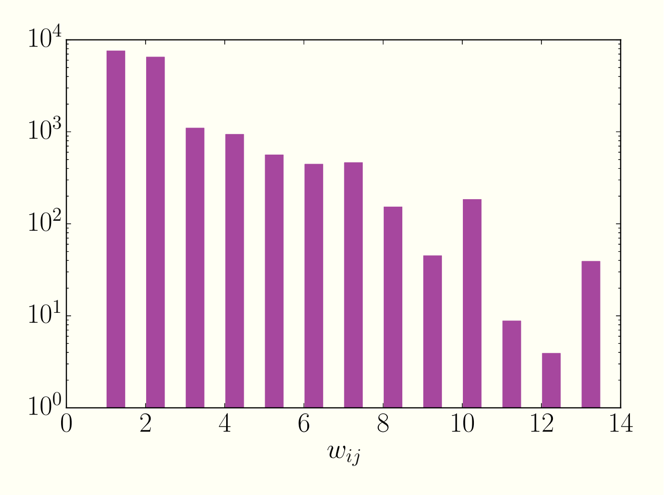


Figure 5.4 – Histogram of edge weights for the WAN network.

5.7.3 Choice of the Static Sensors

When we adopt a static approach, the only degree of freedom we have is the choice of the static sensors. It is known that this choice has an impact on the performance of source localization and that the optimal choice depends on the variance parameter (see Chapter 4. Figure 5.5a compares the performance of KDRS and KMED sensors in terms of the final size of the set of candidate sources $\mathcal{B}$. As in Chapter 4, we observe that for moderate variance, KDRS sensors are better than KMED sensors; for larger ε we have the inverse result. In fact, for large ε, the more uniform placement achieved by KMED can better deal with the noisy observations with respect to the KDRS placement which, instead, enforces the possibility of distinguishing among possible sources in the deterministic case.

We compare KDRS and KMED also for the case where we use dynamic sensors, and we do not restrict the budget for dynamic sensors ($K_d = N - K_s$). As the final set of candidate sources has always cardinality 1, we look instead at the total number of sensors $|\mathcal{U}|$ needed to localize the source. We could think that when we use dynamic sensors and place only a few static sensors ($K_s = 0.02 \cdot N$), the choice of the static sensors has a much smaller effect. Instead, we observe the same result of the static approach. Figures 5.5b and 5.5c show that for both online and offline localization, KDRS emerges as the best choice for static sensors when ε is small but it is outperformed by KMED for larger values of ε.

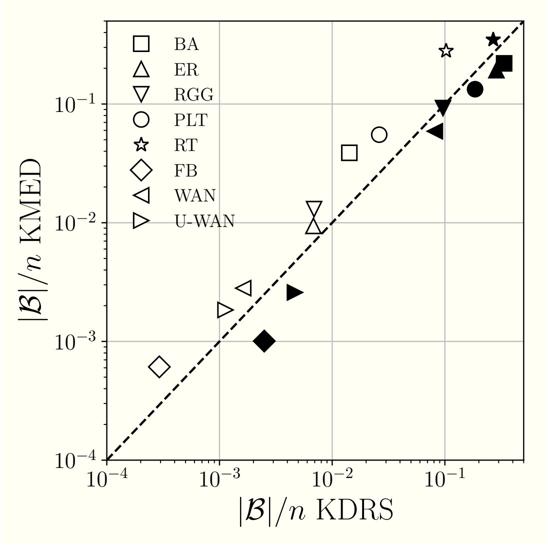
5.7.4 Online vs Offline Localization

Static approach

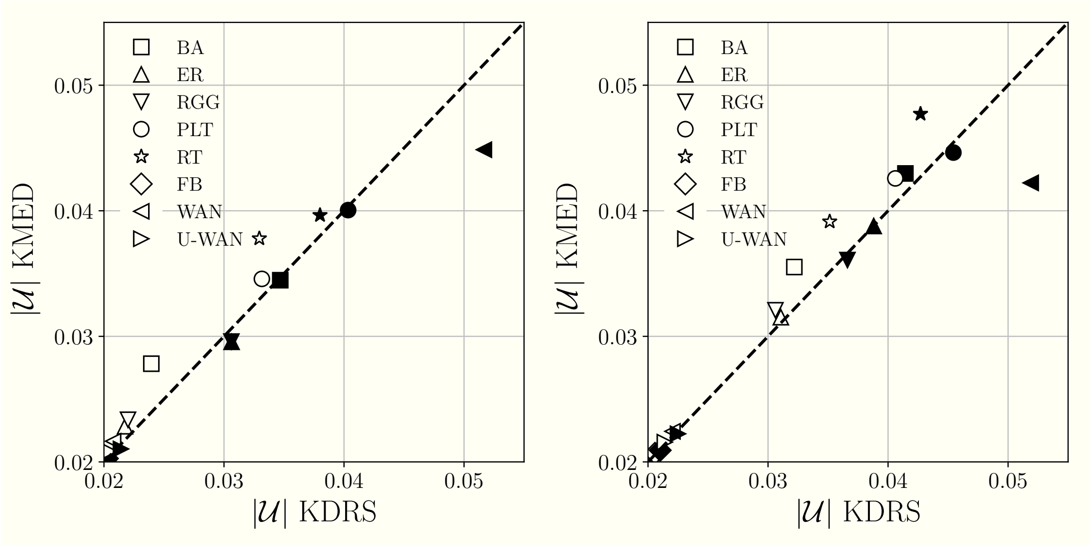
When taking an online approach (S-ON), the computation of $\mathcal{B}$ occurs while the epidemic spreads, hence we can localize the source when many nodes are still not infected. Figure 5.6a (respectively, 5.6b) show the fraction μ of infected nodes when $\mathcal{B}$ contains fewer than 5% of the nodes with KDRS (resp., KMED) sensors. With KDRS sensors, μ is smaller than 60% for all synthetic topologies and smaller than 35% for the real-world networks. As we could expect, with KMED sensors, μ is even smaller (less than 20%), giving an argument for the choice of KMED sensors rather than KDRS sensors also for deterministic epidemics. However, for the tree topologies, the final size of $\mathcal{B}$ can heavily deteriorate when using KMED sensors: For RT, for example, the probability of obtaining $|\mathcal{B}| < 5\% \cdot N$ is lower than 0.05.

Dynamic approach

Also D-ON dramatically reduces, with respect to D-OFF, the fraction of infected nodes at the time when our algorithm terminates. However, in this case, there is a trade-off



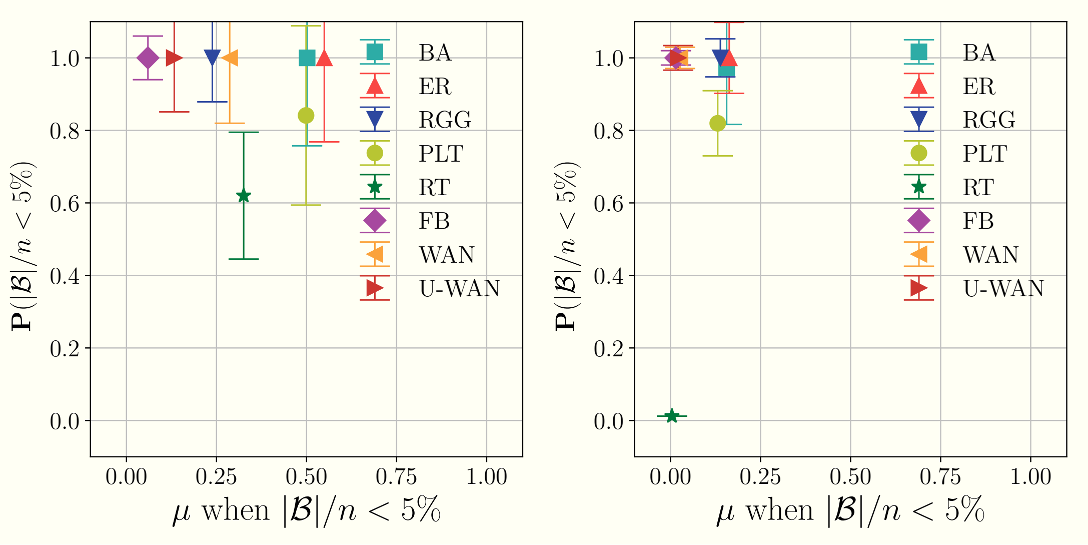
(a) S-OFF & S-ON



(b) $Ks + K_d = N$, D-OFF

(c) $K_s + K_d = N$, D-ON

Figure 5.5 – Comparison of KDRS and KMED for the choice of the static sensors. We use white markers for $\varepsilon = 0$, black for $\varepsilon = 0.2$.



(a) S-ON, $\mathcal{S}$=KDRS

(b) S-ON, $\mathcal{S}$=KMED

Figure 5.6 – S-ON, Fraction μ of infected nodes at the time when $\mathcal{B}$ contains fewer than 5% of the nodes using KDRS sensors (a) and KMED sensors (b). The variance parameter is $\varepsilon = 0$.

between μ and the cost in number of sensors $|\mathcal{U}|$ used for the localization. Figure 5.7 compares the average $|\mathcal{U}|$ for online and offline localization. We see that, independently of the variance parameter and of the network topology, $|\mathcal{U}|$ is smaller for offline localization than for online localization. In fact, in D-OFF, every sensor is already infected by the time it is deployed, hence, when localizing the source offline, we have access to more information.

5.7.5 Static vs Dynamic Localization

Cost of source localization

As stated in Section 5.2, 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 [Chen et al., 2014]. Hence, DMD is a natural benchmark for the cost in terms of number of sensors $|\mathcal{U}|$ of our dynamic approach.

We focus on the deterministic case ($\varepsilon = 0$) with no constraints on the budget for dynamic sensors ($K_d = N - K_s$). We run D-ON and we compare $|\mathcal{U}|/N$ with the (approximate) DMD. The results are depicted in Figure 5.8. For all topologies, $|\mathcal{U}|/N$ is much smaller than DMD$/N$. The improvement is particularly significant for trees whose DMD is very large (equal to the number of leaves [Chen et al., 2014]) but where the topology makes

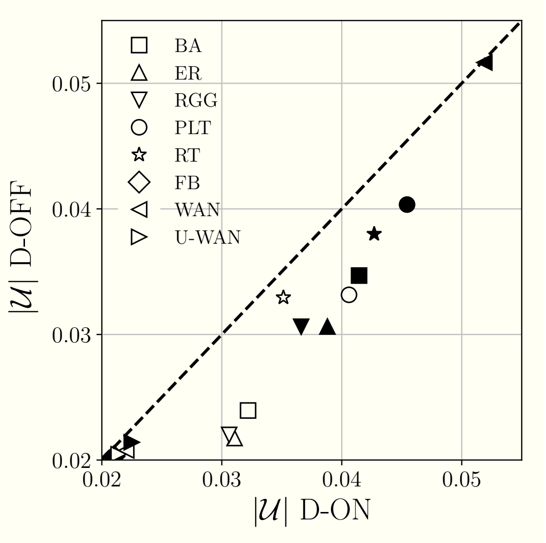


Figure 5.7 – D-ON vs D-OFF: Number of sensors needed to localize the source. We use white markers for $\varepsilon = 0$, black for $\varepsilon = 0.2$. $\mathcal{S}$=KDRS.

it easy for our algorithm to rapidly narrow the search for the source to a small set of candidates. Moreover, we note that $|\mathcal{U}|/N$ is smaller for the real-world topologies than for the synthetic ones and, across all topologies, never exceeds an average of $0.03 \cdot N$, whereas DMD goes up to $0.7 \cdot N$.

Performance with limited budget

We compare the success rate in localizing the source (i.e., the probability $\mathbf{P}(|\mathcal{B}| = 1)$) of a purely-static approach with that of a dynamic approach in both its online and offline variants. We fix a total budget of $K/N = 5\%$ sensors. For the purely-static approach, we have $K_s = K$ and $K_d = 0$; for the dynamic approach we have $K_s = 0.02 \cdot N$ and $K_d = K - K_s = 0.03 \cdot N$. For this experiment, we set $\varepsilon = 0.2$ and we choose KMED static sensors. Figure 5.9a shows that a dynamic approach outperforms a static approach on all topologies. Moreover, in the dynamic case, in line with the results displayed in Figure 5.7, D-OFF achieves a better success rate than D-ON. The only exception is represented by the WAN network: This is probably due to the high variability of the edge weights, which makes it such that sometimes choosing sensors while the epidemic is still local avoids dealing with very noisy transmission times. However, even for this network, D-ON outperforms D-OFF only at an intermediate step in the localization process (i.e., if we stop when $K/N = 0.05$): Looking at Figure 5.7, we see that also for WAN, the total number of sensors needed to localize the source is smaller with D-OFF.

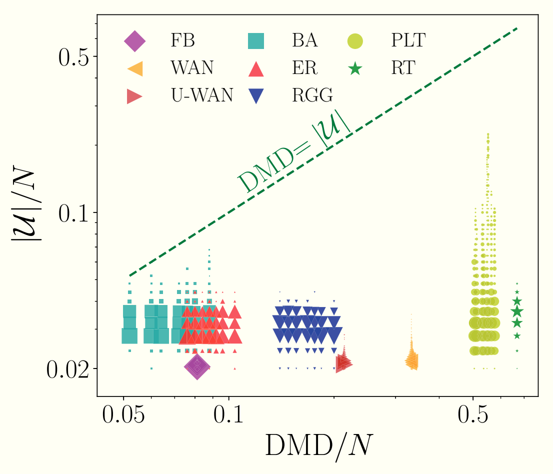
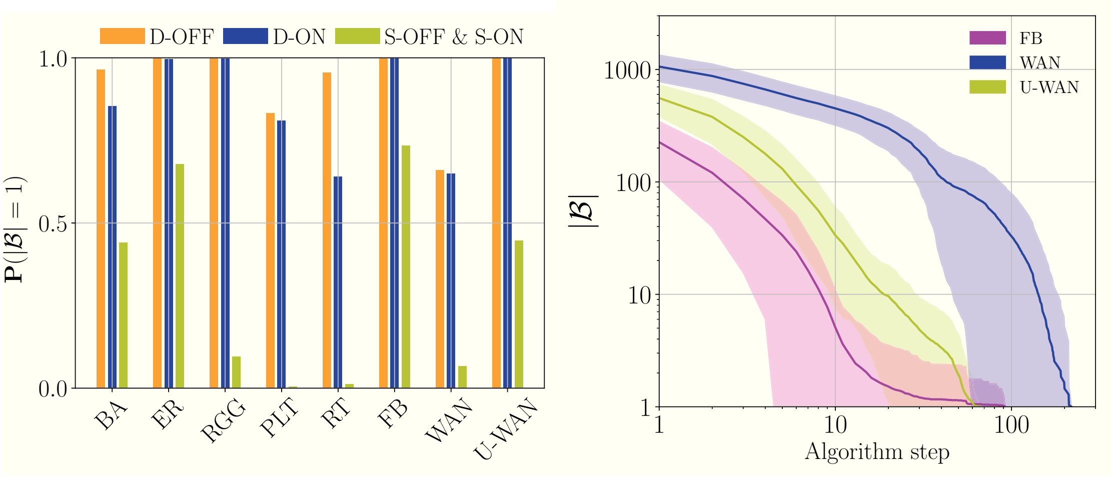


Figure 5.8 – Fraction of sensors needed by D-ON to localize the source compared with the number needed by an optimal offline placement (DMD). Larger markers represent higher concentrations of data points.

5.7.6 Dynamic Sensors: How to Choose, When to Deploy and How Many

Different Gain functions

We study the effect of Gain on the performance of our dynamic algorithm. For each variant, i.e., Size-Gain, DRS-Gain, RC-Gain, we report the relative cost in terms of number of sensors $|\mathcal{U}|/N$ when $K_s + K_d = N$. We experiment with both a deterministic setting ($\varepsilon = 0$) with KDRS static sensors and a non-deterministic setting ($\varepsilon = 0.2$) with KMED static sensors (see Section 5.7.3). The results are depicted in Figure 5.10. We observe that for the real-world networks and $\varepsilon = 0$, all proposed Gain functions have similar performances. For FB and U-WAN, this is true also when $\varepsilon > 0$. These are the cases in which source localization is achieved with the smallest number of sensors. We conclude that, when source localization is less challenging, Gain does not have a strong effect. 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, and in particular for online localization, 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 that have different average infection times (see the definition of DRS-Gain in Eq. (5.12)), does not guarantee that we are able to distinguish them based on the observed infection times [Spinelli et al., 2016]. Instead, as mentioned in Section 5.5, RC-Gain enforces a continuous progress in shrinking the set of candidate sources.



(a) (b)

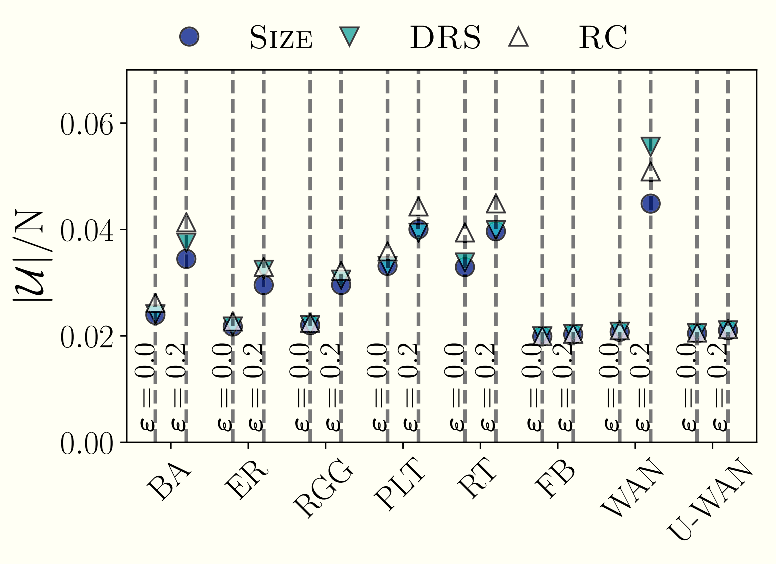
Figure 5.9 – **(a)**: D-ON vs D-OFF vs S-OFF/-ON. Success rate of source localization when $K = K_s + K_d = 0.05 \cdot N$ and $\varepsilon = 0.2$. **(b)**: D-ON. Cardinality of the set $\mathcal{B}$ of candidate sources at successive steps for D-ON (KMED static sensors, $\varepsilon = 0.2$).

Deployment delay

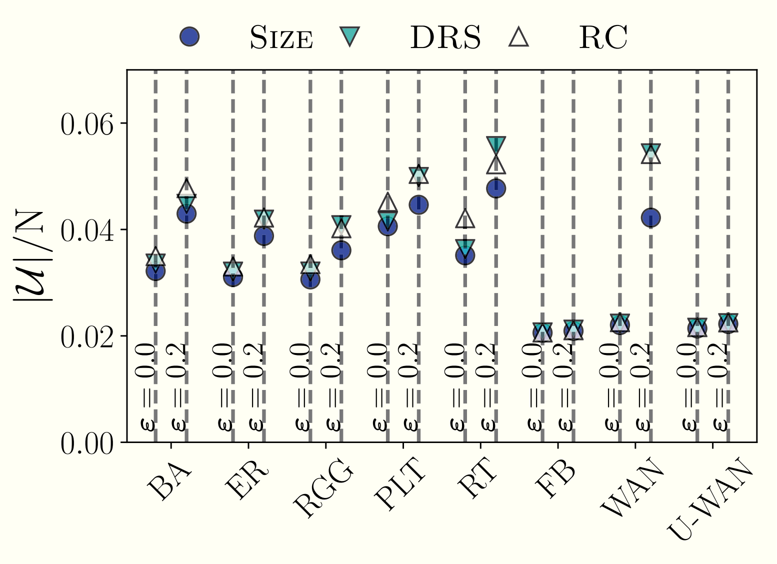
When applying D-ON, an important parameter is the deployment delay θ, i.e., the time between two consecutive placements of a dynamic sensor. On the one hand, the larger θ is, the smaller we expect the cost in terms of number of sensors $|\mathcal{U}|$ to be; on the other hand, the smaller θ is, the less time we expect to need for localizing the source, hence the fewer individuals are infected before we do so. To choose θ, we must also account for the scale of edge weights because, when transmission delays have larger (respectively, smaller) mean, the optimal θ is also likely to be larger (resp., smaller). Here, for simplicity of exposition, we ignore this aspect and experiment only with networks in which all weights are equal to 1. We fix $\varepsilon = 0.2$, we vary θ and look at the number $|\mathcal{D}|$ of dynamic sensors used to localize the source and at the fraction μ of infected individuals at the time of localization. Figures 5.11a and 5.11b display the results for KDRS and KMED sensors, respectively. In both cases, we observe a trade-off between $|\mathcal{D}|$ and μ. When using KDRS sensors, $|\mathcal{D}|$ is smaller, especially for small θ. However, in line with the results of Figure 5.6a, KMED sensors guarantee a smaller fraction of infected μ.

Budget allocation

Given a total budget K, how much of it should we allocate for static and how much for dynamic sensors? We choose KMED static sensors, set $\varepsilon = 0.2$ and run D-ON. We take a total budget $K/N = 5\%$ and we look at the impact of different values of K_s and $K_d = K - K_s$ on the total number of sensors needed to localize the source (which can

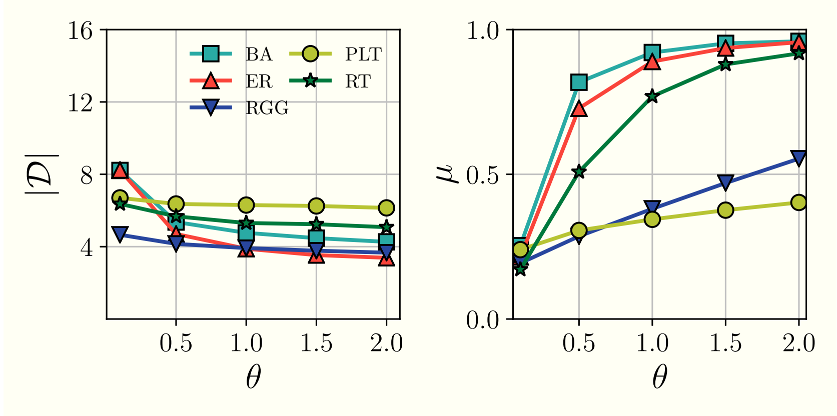


(a) D-OFF

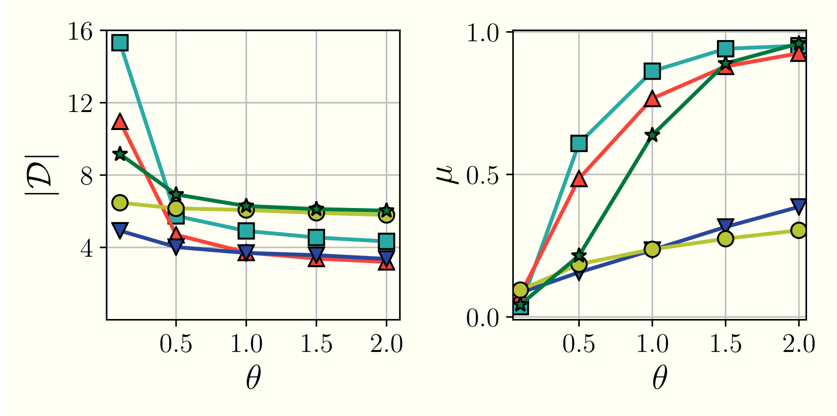


(b) D-ON

Figure 5.10 – Comparison of different GAIN functions.



(a) D-ON, $\mathcal{S}$=KDRS



(b) D-ON, $\mathcal{S}$=KMED

Figure 5.11 – D-ON with varying deployment delay θ: Number of dynamic sensors $|\mathcal{D}|$ needed for source localization and fraction of infected nodes μ at the final stage of the algorithm for varying K_s.

be smaller than K if $|\mathcal{B}| = 1$ is reached before K_d is exhaust). We also evaluate the cardinality of the final set $\mathcal{B}$ and the fraction of infected nodes μ at the final stage of the algorithm. The results are displayed in Figure 5.12. We observe that both $|\mathcal{U}|$ and $|\mathcal{B}|$ increase with K_s, which indicates that a small budget for static sensors guarantees both a lower cost in terms of number of sensors $|\mathcal{U}|$ and a higher precision. However μ is minimized when we use around half of our budget for static sensors. In fact, when K_s is small, we need to deploy many dynamic sensors in order to localize the source; instead, when K_s is large and K_d is small, after placing a few dynamic sensors, we have to wait for the static sensors to get infected in order to use this information to refine $\mathcal{B}$.

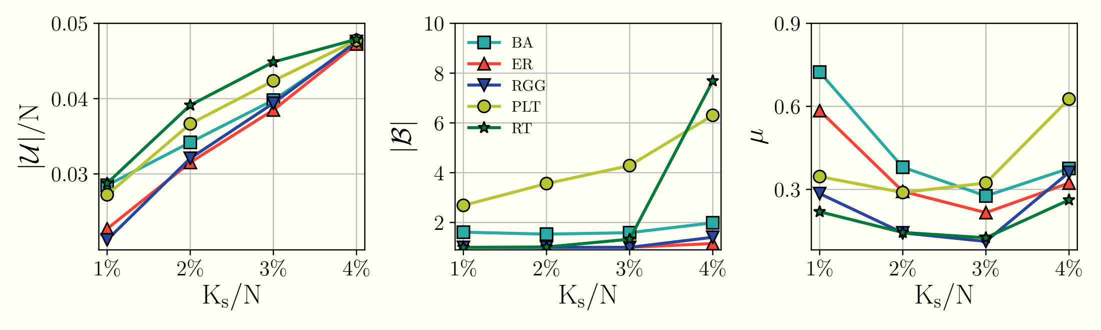
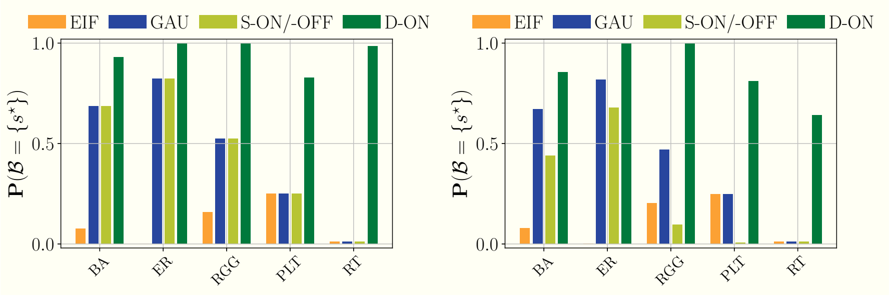


Figure 5.12 – D-ON with constant total budget $K = K_s + K_d = 0.05 \cdot N$: Number of sensors $|\mathcal{U}|$ used by the algorithm, cardinality of the final set of candidate sources $\mathcal{B}$ and fraction of infected nodes μ at the final stage of the algorithm for varying K_s.

5.7.7 Number of Candidate Sources at Successive Steps

We evaluate the runtime of D-ON and D-OFF. The runtime of each iteration of our algorithms is linear in $|\mathcal{B}|$. Hence to estimate the runtime of source localization we look, for the real-world topologies, at how many iterations are needed to localize the source and at how $|\mathcal{B}|$ decreases along the successive iterations of the algorithm. The runtime of D-OFF is smaller than that of D-ON, so we focus on this last case. As we can see in Figure 5.8, 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 (around $0.2 \cdot N$). Hence, source localization is more challenging on the WAN network. This is confirmed by the results shown in Figure 5.9b. In the FB network, with variance parameter $\varepsilon = 0.2$, the source is localized with in average 15 iterations of the localization algorithm. For the U-WAN network, the average number of iterations needed is larger (around 37). 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 number of iterations needed is in average 98) and it is reflected in the average total number of sensors $\mathcal{U}$ used for localization: $0.021 \cdot N$ for FB, $0.022 \cdot N$ for U-WAN and $0.042 \cdot N$ for WAN (see Figure 5.10). This result highlights that the high variability among the edge weights makes source localization substantially more difficult, especially for $\varepsilon > 0$ (see Figure 5.10 for a comparison of the cost in terms of number of sensors $|\mathcal{U}|$ between deterministic and non-deterministic delays). Also the cardinality of $|\mathcal{B}|$ decreases more slowly for the weighted network WAN than for FB and U-WAN. However, for all three topologies, $|\mathcal{B}|$ decreases faster than linearly (note the logarithmic scale in Figure 5.9b), confirming the feasibility of our approach to source localization for a variety of real-world networks.



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

Figure 5.13 – Comparison of D-ON and S-ON/-OFF with the baseline methods GAU and EIF.

5.7.8 Comparison with Existing Methods

We compare our algorithms for source localization with the two following existing methods:

- GAU: this method estimates the source through maximum-likelihood assuming Gaussian transmission delays. It was initially proposed by Pinto et al. [2012] under the assumption that each sensor reveals its infection time and from which node it received the infection. In Chapter 4 we generalized this estimator to the setting where the information about who-infected-whom is not available; we also showed that the same method can be applied with different infection-delays distributions (such as truncated-Gaussian and uniform). Here, we consider the improved estimator used in Chapter 4.

- EIF: this method estimates the source by computing, for every candidate source v, an infection tree $\mathcal{T}_v$ rooted at v that is compatible with the observed infection times. Every spreading tree $\mathcal{T}_v$ is given a cost that quantifies how much the observed infection times deviate from the expected infection times when the infection spreads along $\mathcal{T}_v$. The estimated source is the root v of the spreading tree $\mathcal{T}_v$ with minimal cost. The method was proposed by Zhu et al. [2015] and it is independent from the transmission-delay distribution, only the average delays are needed for the estimation.

We take $K = 5\%N$. For GAU, EIF and S-ON/-OFF, all sensors are static; for D-ON $K_s = 2\%N$ sensors are static and $K_d = K - K_s = 3\%N$ sensors are dynamic. We look at $\mathbf{P}(\mathcal{B} = \{s^*\})$, i.e., the probability that the source is correctly identified without ties, for $\varepsilon = 0$ (Figure 5.13a) and for $\varepsilon = 0.2$ (Figure 5.13b). For $\varepsilon = 0$, the performance of GAU and S-ON/-OFF is identical: They both identify the correct equivalence class of the source (see Definition 1.3) but cannot distinguish among nodes in the same equivalence class. For tree networks (PLT and RT) the performance of EIF is also identical to that of S-ON/-OFF and GAU: On trees, all the nodes in the equivalence class of the source, and only these nodes, are the roots of the minimal cost spreading trees. Instead, on non-tree networks, the presence of loops makes it more challenging to correctly identify the spreading tree and the performance of EIF is poorer than that of GAU and S-ON/-OFF. For all network topologies, D-ON beats all other methods by a large margin. For $\varepsilon \neq 0$, the performance of S-ON/-OFF is lower than that of GAU. In fact, our methods are designed to have maximum recall ($\mathbf{P}(s^\star \in \mathcal{B}) = 1$) and, for both S-ON/-OFF and D-ON, all nodes that have a positive probability of being the source are contained in $\mathcal{B}$. When moving from a deterministic to a non-deterministic setting, the various methods suffer a performance drop in different ways: S-ON/-OFF and D-ON have a drop in precision; GAU and EIF have a drop in recall. In a non-deterministic setting, D-ON still detects the source with no ambiguity with a high probability, again strongly outperforming the alternative methods.

5.7.9 Resistance to Unbounded Delay Distributions

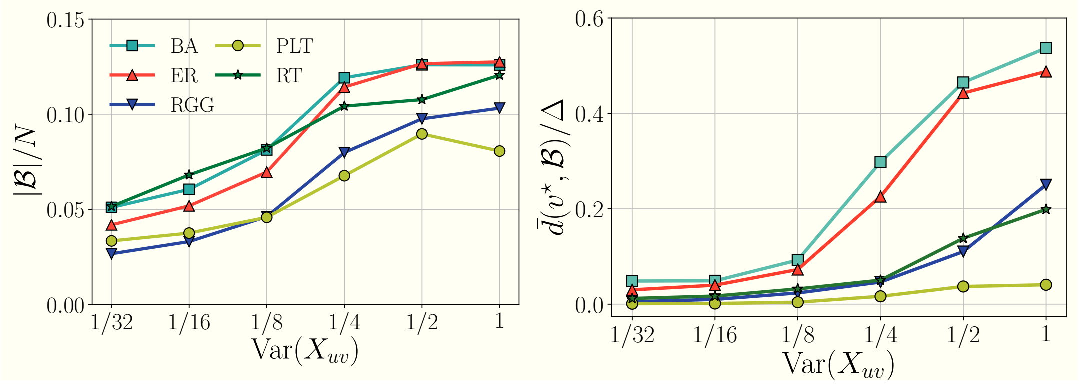
Our theoretical results are derived under the hypothesis of bounded-support for the distribution of the transmission delays (see Section 5.3-5.6). However, in some applications, we work with transmission delays that are not upper bounded by a constant. We test our D-ON method when each transmission delay X_{uv} is drawn from a Gamma distribution $\Gamma(k, 1/k)$ (hence $\mathbb{E}[X_{uv}] = 1$ and $Var(X_{uv}) = 1/k$) that includes the case of exponential transmission delays ($k = 1$). In this setting, D-ON is not guaranteed to always detect the source (i.e., Theorem 5.15 does not hold) and it can happen that $v^\star$ is removed from the set of candidates $\mathcal{B}$.

Let ε_0 be the minimum value such that $\mathbf{P}(X_{uv} \in [w_{uv}(1-\varepsilon_0), w_{uv}(1+\varepsilon_0)]) = 0.75$. In order to account for the variance of X_{uv} but still enforce the removal of nodes from $\mathcal{B}$ we run D-ON with $\varepsilon = \min(\varepsilon_0(k), 0.6)$. Figure 5.14a shows the final size of the set $\mathcal{B}$, and Figure 5.14b depicts the average distance $\bar{d}$ from $s^\star$ to the nodes in $\mathcal{B}$. For moderate variance, $\bar{d}$ is very small for all the topologies considered; for PLT, RGG and RT this distance is very small even for large values of the variance, indicating a good performance of our methods, especially for tree networks, even with unbounded and highly variable transmission delays.

5.8 Additional Technical Details

5.8.1 S-OFF

We give the technical details regarding the computation of the set of candidate sources $\mathcal{B}$, starting from the case of deterministic epidemics and later extending our results to the more general case when $\varepsilon > 0$.



(a) (b)

Figure 5.14 – Performance of D-ON when the transmission delays are Gamma random variables with mean 1 in terms of (a) $\mathbf{P}(\mathcal{B} = \{v^\star\})$ and (b) average distance $\bar{d}(v^\star, \mathcal{B})$ from the nodes in $\mathcal{B}$ to $v^\star$ (rescaled with the diameter Δ). $K_d = 10\%$ and ε is chosen as described in Section 5.7.9.

We first note that, using Definition 5.3 we can rewrite $\mathbf{P}(\mathcal{O}|v^\star = v)$ as

$$\mathbf{P}(\mathcal{O}|v^\star = v) = \mathbf{P}\left(\bigcap_{\omega_i \neq \omega_j \in \mathcal{O}} A_{\omega_i,\omega_j} \middle| v^\star = v \right). \tag{5.14}$$

The next lemma formalizes that, when epidemics spread deterministically, the only source of randomness in the epidemic is the value of $v^\star$.

Lemma 5.16. *Let $\mathcal{O}$ be a set of observations and let $\varepsilon = 0$. Then, for all $v \in V$, $\mathbf{P}(\mathcal{O}|v^\star = v) \in \{0, 1\}$.*

Proof. Let us pick $v \in V$ such that $\mathbf{P}(\mathcal{O}|v^\star = v) > 0$. We want to prove that $\mathbf{P}(\mathcal{O}|v^\star = v) = 1$. We have

$$\begin{aligned}
&\mathbf{P}(\mathcal{O}|v^\star = v) > 0 \\
&\Leftrightarrow \mathbf{P}\left(\bigcap_{\omega_i \neq \omega_j \in \mathcal{O}} A_{\omega_i,\omega_j} | v^\star = v \right) > 0 \\
&\Rightarrow \mathbf{P}(A_{\omega_i,\omega_j} | v^\star = v) > 0 \;\; \forall \;\; \omega_i \neq \omega_j \in \mathcal{O} \\
&\Leftrightarrow \mathbf{P}(T(v, u_i) - T(v, u_j) = t_i - t_j) > 0 \;\; \forall \;\; \omega_i \neq \omega_j \in \mathcal{O} \\
&\overset{(a)}{\Leftrightarrow} \mathbf{P}(A_{\omega_i,\omega_j} | v^\star = v) = 1 \;\; \forall \;\; \omega_i \neq \omega_j \in \mathcal{O} \\
&\Leftrightarrow \mathbf{P}(\mathcal{O}|v^\star = v) = 1,
\end{aligned}$$

where (a) holds because $T(v, u_i) - T(v, u_j)$ is deterministic and equal to $d(v, u_i) - d(v, u_j)$. ☐

A particular case of Lemma 5.16 is the following.

Lemma 5.17. *Let* $\varepsilon = 0$ *and let* $\omega_1 \triangleq (u_1, t_1)$ *and* $\omega_2 \triangleq (u_2, t_2)$ *be two observations. Then* $\mathbf{P}(A_{\omega_1,\omega_2} | v^\star = v) > 0$ *if and only if* $\mathbf{P}(A_{\omega_1,\omega_2} | v^\star = v) = 1$*. Moreover,* $\mathbf{P}(A_{\omega_1,\omega_2} | v^\star = v) > 0$ *if and only if* $d(v, u_1) - d(v, u_2) = t_1 - t_2$*.*

Proof. As in the proof of Lemma 5.16, $\mathbf{P}(A_{\omega_1,\omega_2} | v^\star = v) > 0 \Leftrightarrow \mathbf{P}(T(v, u_1) - T(v, u_2) = t_1 - t_2) > 0 \Leftrightarrow \mathbf{P}(T(v, u_1) - T(v, u_2) = t_1 - t_2) = 1 \Leftrightarrow d(v, u_1) - d(v, u_2) = t_1 - t_2$. □

We are now ready to prove Proposition 5.4 which gives a practical way of computing $\mathcal{B}$.

Proposition 5.4. *Let* $\mathcal{O}$ *be a set of observations and let* $\varepsilon = 0$*. Let* $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}$ *be a fixed observation, which we call the reference observation. Then, the set of candidate sources* $\mathcal{B}$ *is*

$$\mathcal{B} = \bigcap_{\omega \in \mathcal{O} \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}. \tag{5.3}$$

Proof. We have

$$\begin{aligned} v \in \mathcal{B} &\overset{(a)}{\Leftrightarrow} \mathbf{P}(\mathcal{O} | v^\star = v) = 1 \\ &\overset{(b)}{\Leftrightarrow} \mathbf{P}(A_{\omega_i,\omega_j} | v^\star = v) = 1 \;\; \forall \omega_i \neq \omega_j \in \mathcal{O} \\ &\overset{(c)}{\Leftrightarrow} v \in \mathcal{B}_{\omega_i,\omega_j} \;\; \forall \omega_i \neq \omega_j \;\; \in \mathcal{O} \\ &\Leftrightarrow v \in \bigcap_{\omega_i \neq \omega_j \in \mathcal{O}} \mathcal{B}_{\omega_i,\omega_j}, \end{aligned}$$

where (a) holds by Lemma 5.16, (b) follows from (5.14) and (c) holds by Lemma 5.17.

To prove $\mathcal{B} \subseteq \bigcap_{\omega \in \mathcal{O} \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}$ it is enough to note that $\bigcap_{\omega_i \neq \omega_j \in \mathcal{O}} \mathcal{B}_{\omega_i,\omega_j} \subseteq \bigcap_{\omega \in \mathcal{O} \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}$. For the reverse inclusion, take $v \in \bigcap_{\omega \in \mathcal{O} \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}$ and $\omega_i \triangleq (u_i, t_{u_i}), \omega_j \triangleq (u_j, t_{u_j}) \in \mathcal{O} \setminus \{\omega_1\}$, $\omega_i \neq \omega_j$. Since $v \in \mathcal{B}_{\omega_1,\omega_i} \cap \mathcal{B}_{\omega_1,\omega_j}$, by Lemma 5.17 we have

$$d(v, u_i) - d(v, u_1) = t_i - t_1, \tag{5.15}$$

$$d(v, u_j) - d(v, u_1) = t_j - t_1. \tag{5.16}$$

By subtracting (5.15) from (5.16), we get $d(v, u_i) - d(v, u_j) = t_i - t_j$, which, again by Lemma 5.17 implies $v \in \mathcal{B}_{\omega_i,\omega_j}$. Hence we can conclude that $v \in \bigcap_{\omega_i \neq \omega_j \in \mathcal{O}} \mathcal{B}_{\omega_i,\omega_j}$. □

We now turn to non-deterministic epidemics and give a proof of Proposition 5.5 which is at the basis of the definition of the superset $\widetilde{\mathcal{B}}$ of candidate sources (see (5.5)).

Proposition 5.5. *Let* $0 < \varepsilon < 1$*, let* $\omega_1 \triangleq (u_1, t_{u_1})$*,* $\omega_2 \triangleq (u_2, t_{u_2}) \in \mathcal{O}$*,* $\omega_1 \neq \omega_2$*, and let* $v \in \mathcal{B}$*. Then*

$$|d(v, u_1) - d(v, u_2) - t_{u_1} + t_{u_2}| \leq \varepsilon(d(v, u_1) + d(v, u_2)). \tag{5.4}$$

Proof. Since $v \in \mathcal{B}$, $\mathbf{P}(A_{\omega_1,\omega_2}|v^\star = v) > 0$. We prove that, if $v^\star = v$, (5.4) holds. From this we can conclude that (5.4) holds for every $v \in \mathcal{B}$, because if there were $v \in \mathcal{B}$ such that (5.4) does not hold, we would have $\mathbf{P}(A_{\omega_1,\omega_2}|v^\star = v) = 0$, giving a contradiction with $v \in \mathcal{B}$.

Recall that, if $v^\star = v$, the infection time of u is $t_u = t^\star + T(v, u)$. Since the transmission delay along edge xy has range $[(1 - \varepsilon)w_{xy}, (1 + \varepsilon)w_{xy}]$, we have

$$T(v, u) \leq (1 + \varepsilon)d(v, u). \tag{5.17}$$

If $\mathcal{Q}$ is the collection of all paths connecting v and u and, for $p \in \mathcal{Q}$, $d_p(v, u)$ is the weighted length of path p we have

$$T(v, u) \geq (1 - \varepsilon) \min_{p \in \mathcal{Q}} d_p(v, u) = (1 - \varepsilon)d(v, u). \tag{5.18}$$

Combining inequalities (5.17) and (5.18) we obtain

$$|T(v, u_1) - d(v, u_1)| \leq \varepsilon d(v, u_1), \tag{5.19}$$

$$|T(v, u_2) - d(v, u_2)| \leq \varepsilon d(v, u_2). \tag{5.20}$$

From (5.19) and (5.20) and using the relation $T(v, u_1) - T(v, u_2) = t_{u_1} - t_{u_2}$ we obtain (5.4). □

Algoritm 9 gives the pseudo-code for computing of $\widetilde{\mathcal{B}}$.

Algorithm 9 S-OFF - non-deterministic epidemic

```
Require: $\mathcal{O}$ set of observations
  $\widetilde{\mathcal{B}} \leftarrow V$
  for $(u, t_u), (z, t_z) \in \mathcal{O}$, $u \neq z$ do
      for $v \in \widetilde{\mathcal{B}}$ do
          $D \leftarrow |d(v, u) - d(v, z) - t_u + t_z|$
          $E \leftarrow \varepsilon(d(v, u) + d(v, z))$
          if $D > E$ then
              remove $v$ from $\widetilde{\mathcal{B}}$
  return $\widetilde{\mathcal{B}}$
```

Finally, we give a proof of Proposition 5.8.

Proposition 5.8. *Let $\mathcal{U}$ be the sensor set. Let*

$$\Delta \triangleq \max_{u \in \mathcal{U}, v \in V} d(v, u)$$

and

$$\phi \triangleq \min_{[v_1]_{\mathcal{U}} \neq [v_2]_{\mathcal{U}}} \max_{u_1, u_2 \in \mathcal{U}} |d(v_1, u_1) - d(v_1, u_2) - d(v_2, u_1) + d(v_2, u_2)|.$$

If $\varepsilon < \varepsilon_0 \triangleq \phi/4\Delta$ and $v^\star = v$, then $\widetilde{\mathcal{B}} \subseteq [v]_{\mathcal{U}}$.

Proof. Let $v^\star = v$ and $w \notin [v]_{\mathcal{U}}$. We want to prove that $w \notin \widetilde{\mathcal{B}}$. By hypothesis, there exist $u_1, u_2 \in \mathcal{U}$ such that

$$|d(v, u_1) - d(v, u_2) - d(w, u_1) + d(w, u_2)| \geq \phi. \tag{5.21}$$

For every $z \in V$, let $\mu_z(u_1, u_2) \triangleq d(z, u_2) - d(z, u_1)$. By Equation (5.4), the deviation of $t_{u_2} - t_{u_1}$ from $\mu_v(u_1, u_2)$ is upper bounded by

$$|t_{u_2} - t_{u_1} - \mu_v(u_1, u_2)| \leq \varepsilon(d(v, u_2) + d(v, u_1)) \leq 2\varepsilon\Delta.$$

Moreover, by definition of $\widetilde{\mathcal{B}}$, a similar bound holds for every $z \in \widetilde{\mathcal{B}}$:

$$|t_{u_2} - t_{u_1} - \mu_z(u_1, u_2)| \leq \varepsilon(d(z, u_2) + d(z, u_1)) \leq 2\varepsilon\Delta.$$

Assume by contradiction that $w \in \widetilde{\mathcal{B}}$. Then, by applying the triangle inequality and the hypothesis $\varepsilon < \phi/4\Delta$ we have

$$\begin{aligned} |\mu_v(u_1, u_2) - \mu_w(u_1, u_2)| &\leq |t_{u_2} - t_{u_1} - \mu_v(u_1, u_2)| + |t_{u_2} - t_{u_1} - \mu_w(u_1, u_2)| \\ &\leq 4\varepsilon\Delta < \phi \end{aligned} \tag{5.22}$$

which contradicts (5.21). Hence, $\widetilde{\mathcal{B}} \subseteq [v]_{\mathcal{U}}$. □

5.8.2 S-ON

We show how $\mathcal{B}$ and $\widetilde{\mathcal{B}}$ can be updated when we have negative observations.

Similarly to Section 5.8.1, using Definitions 5.3 and 5.9 we can rewrite $\mathbf{P}(\mathcal{O}_t | v^\star = v)$ as

$$\mathbf{P}\left(\left(\bigcap_{\omega_i \neq \omega_j \in \mathcal{O}_t^+} A_{\omega_i, \omega_j}\right) \bigcap \left(\bigcap_{\substack{\omega_i \in \mathcal{O}_t^+, \\ \omega_j \in \mathcal{O}_t^-}} A_{\omega_i, \omega_j}^t\right) \middle| v^\star = v\right).$$

The next lemma extends Lemma 5.16 and Lemma 5.17 to the case in which $\mathcal{O}$ contains negative observations.

Lemma 5.18. *Let* $t \in \mathbb{R}$, $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}_t^+$ *and* $\omega_2 \triangleq (u_2, \emptyset) \in \mathcal{O}_t^-$. *Then* $\mathbf{P}(A^t_{\omega_1,\omega_2} | v^\star = v) > 0$ *if and only if* $\mathbf{P}(A^t_{\omega_1,\omega_2} | v^\star = v) = 1$ *and* $d(v, u_1) - d(v, u_2) < t_{u_1} - t$.

Proof. We have the following sequence of equivalences.

$$\begin{aligned}
&\mathbf{P}(A^t_{\omega_1,\omega_2} | v^\star = v) > 0 \\
&\Leftrightarrow \mathbf{P}(T(v^\star, u_1) - T(v^\star, u_2) < t_{u_1} - t) > 0 \\
&\overset{(a)}{\Leftrightarrow} \mathbf{P}(T(v^\star, u_1) - T(v^\star, u_2) < t_{u_1} - t) = 1 \\
&\Leftrightarrow d(v^\star, u_1) - d(v^\star, u_2) < t_{u_1} - t
\end{aligned} \tag{5.23}$$

where (a) holds because, given the value of $v^\star$, $T(v^\star, u_1) - T(v^\star, u_2)$ is deterministic and equal to $d(v^\star, u_1) - d(v^\star, u_2)$. ☐

Proposition 5.10. *Let* $t \in \mathbb{R}$, $\mathcal{O}_t$ *be the set of observations at time* t *and* $\varepsilon = 0$. *Let* $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}^+_{\tau^\star}$ *be the first positive observation that we call the reference observation. Then, the set of candidate sources* $\mathcal{B}_t$ *is*

$$\mathcal{B}_t = \left(\bigcap_{\omega \in \mathcal{O}_t^+ \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega} \right) \bigcap \left(\bigcap_{\omega \in \mathcal{O}_t^-} \mathcal{B}_{\omega_1,\omega,t} \right).$$

Moreover, if $t, t' \in \mathbb{R}$, $t' > t$, $\mathcal{B}_{t'} \subseteq \mathcal{B}_t$.

Proof. The fact that $\mathcal{B}_t$ contains all and only the nodes that have a positive probability to be the source given the information available at time t is a direct consequence of the definition of $\mathcal{B}_t$ and $\mathcal{O}_t$.

Similarly to the proof of Proposition 5.4 we have

$$v \in \mathcal{B}_t \Leftrightarrow v \in \left(\bigcap_{\omega_i \neq \omega_j \in \mathcal{O}_t^+} \mathcal{B}_{\omega_i,\omega_j} \right) \bigcap \left(\bigcap_{\omega_i \in \mathcal{O}_t^+, \omega_j \in \mathcal{O}_t^-} \mathcal{B}_{\omega_i,\omega_j,t} \right)$$

and hence

$$\mathcal{B}_t \subseteq \left(\bigcap_{\omega \in \mathcal{O}_t^+ \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega} \right) \bigcap \left(\bigcap_{\omega \in \mathcal{O}_t^-} \mathcal{B}_{\omega_1,\omega,t} \right).$$

If $\omega_i, \omega_j \in \mathcal{O}_t^+ \setminus \{\omega_1\}$, $\omega_i \neq \omega_j$, as in the proof of Proposition 5.4 we have that $v \in \mathcal{B}_{\omega_1,\omega_i} \cap \in \mathcal{B}_{\omega_1,\omega_j}$ implies $v \in \mathcal{B}_{\omega_i,\omega_j}$. Let now $\omega_i \triangleq (u_i, t_{u_i}) \in \mathcal{O}_t^+ \setminus \{\omega_1\}$ and $\omega_j \triangleq (u_j, \emptyset) \in \mathcal{O}_t^-$ and take $v \in \mathcal{B}_{\omega_1,\omega_i} \cap \mathcal{B}_{\omega_j,\omega_1,t}$. By Lemma 5.17 and Lemma 5.18 we have

$$d(u_i, v) - d(u_1, v) = t_{u_i} - t_{u_1} \tag{5.24}$$

$$d(u_1, v) - d(u_j, v) < t_{u_1} - t. \tag{5.25}$$

Combining (5.24) and (5.25), we have $d(u_i, v) - d(u_j, v) < t_{u_i} - t$ and, by Lemma 5.18, $v \in \mathcal{B}_{\omega_i,\omega_j,t}$. Hence we proved

$$\mathcal{B}_t \supseteq \Big(\bigcap_{\omega \in \mathcal{O}_t^+ \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega} \Big) \bigcap \Big(\bigcap_{\omega \in \mathcal{O}_t^-} \mathcal{B}_{\omega_1,\omega,t} \Big).$$

Let now $t, t' \in \mathbb{R}$ and $t' > t$ and take $v \in \mathcal{B}_{t'}$. As $\mathcal{O}_{t'}^+ \supseteq \mathcal{O}_t^+$,

$$v \in \bigcap_{\omega_i \in \mathcal{O}_t^+ \setminus \{\omega_1\}} \mathcal{B}_{\omega_1,\omega}.$$

If $\omega_j \triangleq (u_j, \emptyset) \in \mathcal{O}_{t'}^-$, by Lemma 5.18, $d(u_1, v) - d(u_j, v) < t_{u_1} - t'$ and, since $t' > t$, $d(u_1, v) - d(u_j, v) < t_{u_1} - t$. Hence, again by Lemma 5.18, $v \in \mathcal{B}_{\omega_1,\omega_j,t}$ and $\mathcal{B}_{t'} \subseteq \mathcal{B}_t$. □

We now turn to non-deterministic epidemics. As in S-OFF, when $0 < \varepsilon < 1$, we compute a superset of the candidate set $\widetilde{\mathcal{B}}_t \supseteq \mathcal{B}_t$. To do this, we extend Proposition 5.5 to account for negative observations.

Proposition 5.19. *Let* $t \geq \tau^\star$, $\omega_1 \triangleq (u_1, t_{u_1}) \in \mathcal{O}_t^+$, $\omega_2 \triangleq (u_2, \emptyset) \in \mathcal{O}_t^-$ *and let* $v \in \mathcal{B}_t$. *Then,*

$$d(u_1, v) - d(u_2, v) - t_{u_1} + t < \varepsilon(d(u_1, v) + d(u_2, v)). \tag{5.26}$$

Proof. The proof of the result follows closely that of Proposition 5.5. We limit ourselves to highlighting the differences. If $v^\star = v$, we have

$$T(v, u_1) \geq d(v, u_1) - \varepsilon d(v, u_1), \tag{5.27}$$

$$T(v, u_2) \leq d(v, u_2) + \varepsilon d(v, u_2). \tag{5.28}$$

Combining (5.27) and (5.28) and using the relation $T(v, u_1) - T(v, u_2) < t_{u_1} - t$, we obtain (5.26). □

In view of Proposition 5.19 and Remark 5.6, we can compute and update $\widetilde{\mathcal{B}}$ with Algorithm 10. Like for Algorithm 9, the runtime of Algorithm 10 is $O(K_s^2 N)$.

Algorithm 10 S-ON - non-deterministic epidemic

```
Require: Observation sets $\{\mathcal{O}_i^+\}_{i=1}^F$, $\{\mathcal{O}_i^-\}_{i=1}^F$
  $\widetilde{\mathcal{B}}_0 \leftarrow V$
  $i \leftarrow 1$
  while $i \leq F$ and $|\widetilde{\mathcal{B}}_{i-1}| > 1$ do
    $i \leftarrow i + 1$
    $\widetilde{\mathcal{B}}_i \leftarrow \widetilde{\mathcal{B}}_{i-1}$
    for $(u, t_u) \in \mathcal{O}_i^+ \backslash \mathcal{O}_{i-1}^+, (z, t_z) \in \mathcal{O}_i^+$ do
      for $v \in \widetilde{\mathcal{B}}_i$ do
        $D \leftarrow |d(u, v) - d(z, v) - t_u + t_z|$
        $E \leftarrow \varepsilon(d(u, v) + d(u, v))$
        if $D > E$ then
          remove $v$ from $\widetilde{\mathcal{B}}_i$
    for $(u, \emptyset) \in \mathcal{O}_i^-, (z, t_z) \in \mathcal{O}_i^+$ do
      for $v \in \widetilde{\mathcal{B}}_i$ do
        $D \leftarrow d(z, v) - d(u, v) - t_z + t_i$
        $E \leftarrow \varepsilon(d(u, v) + d(z, v))$
        if $D \geq E$ then
          remove $v$ from $\widetilde{\mathcal{B}}_i$
  return $\mathcal{B}_i$
```

5.8.3 D-ON

We give here some details concerning our D-ON algorithm.

As a pseudo-code for the complete algorithm would be quite involved (hence not very helpful for the reader), we limit ourselves to giving the pseudo-code for the subroutines with which, at a time t, we update the candidate set $\mathcal{B}$, the sensor set $\mathcal{U}$, and the observation set $\mathcal{O} = \mathcal{O}^+ \cup \mathcal{O}^-$. The initialization, i.e., the first computation of $\mathcal{B}$ at time $\tau^\star$ is done in D-ON as for S-ON (i.e., as in the first iteration of the **while** loop of Algorithm 7).

At time t, the candidate set $\mathcal{B}$, the sensor set $\mathcal{U}$, and the observation set $\mathcal{O} = \mathcal{O}^+ \cup \mathcal{O}^-$ are updated in two cases:

I) if $t = \tau^\star + \theta j$, $j \in \mathbb{N}$, i.e., at time t a new dynamic sensor is added. In this case $\mathcal{B}$, $\mathcal{U}$ and $\mathcal{O} = \mathcal{O}^+ \cup \mathcal{O}^-$ are updated with the subroutine presented in Algorithm 11.

II) if $t = t_u > \tau^\star$, i.e., t is the infection time of a static sensor or of a node that was chosen as dynamic sensor before time t but was not yet infected at time t. In this case $\mathcal{B}$, $\mathcal{U}$ and $\mathcal{O} = \mathcal{O}^+ \cup \mathcal{O}^-$ are updated with the subroutine presented in Algorithm 12.

In Algorithm 11 and 12, $\bar{t}$ denotes the time at which $\mathcal{B}$, $\mathcal{U}$ and $\mathcal{O}$ where last updated before time t. If t is the time of the first update, $\bar{t} = \tau^\star$. To simplify the notation, the time index for the set $\mathcal{B}$ is omitted.

Algorithm 11 D-ON - Update I - deterministic epidemic

```
Require: $\mathcal{B}, \mathcal{U}, \mathcal{O}_{\bar{t}}, \omega_1 \triangleq (u_1, t_1) \in \mathcal{O}^+_{\tau^\star}$
  $d' \leftarrow \mathrm{argmax}_{d \in V \backslash \mathcal{U}} \textsc{Gain}_{\mathcal{U}}(d)$
  $\mathcal{U} \leftarrow \mathcal{U} \cup \{d'\}$
  if $d'$ is infected then
    $t_{d'} \leftarrow$ infection time of $d'$
    $\mathcal{O}^+_t \leftarrow \mathcal{O}^+_{\bar{t}} \cup (d', t_{d'})$
    $\mathcal{O}^-_t \leftarrow \mathcal{O}^-_{\bar{t}}$
    for $v \in \mathcal{B}$ do
      if $d(d', v) - d(u_1, v) \neq t_{d'} - t_1$ then
        remove $v$ from $\mathcal{B}$
    for $\omega \triangleq (u, \emptyset) \in \mathcal{O}^-_t$ do
      for $v \in \mathcal{B}$ do
        if $d(u, v) - d(u_1, v) < t - t_1$ then
          remove $v$ from $\mathcal{B}$
  else
    $\mathcal{O}^+_t \leftarrow \mathcal{O}^+_{\bar{t}}$
    $\mathcal{O}^-_t \leftarrow \mathcal{O}^-_{\bar{t}} \cup (d', \emptyset)$
    for $\omega \triangleq (u, \emptyset) \in \mathcal{O}^-_t$ do
      for $v \in \mathcal{B}$ do
        if $d(u, v) - d(u_1, v) < t - t_1$ then
          remove $v$ from $\mathcal{B}$
```

The extensions of Algorithm 11 and 12 to non-deterministic epidemics follow from Proposition 5.19 and Algorithm 10.

5.8.4 Extending the Gain Functions to Negative Observations

In online source localization, dynamic sensors can yield negative observations. For this reason, the computation of $g^{\mathrm{SIZE}}_{\mathcal{U}}$ and $g^{\mathrm{DRS}}_{\mathcal{U}}$ given in Section 5.5.2 should slightly change to account for the case in which a dynamic sensor is not infected by the time at which it is deployed.

Definition 5.20 (Possible infection times). *Let $\mathcal{U}$ be a set of sensors, $c \in V \backslash \mathcal{U}$, $\mathcal{O}_{\mathcal{U}} \triangleq \{(u, t_u), u \in \mathcal{U}\}$ and fix $(u_1, t_1) \in \mathcal{O}_{\mathcal{U}}$ arbitrarily. Let $\mathcal{B}_{\mathcal{U}}$ be the set of candidate sources after observing the infection times of the nodes in $\mathcal{U}$, i.e., $\mathcal{B}_{\mathcal{U}} = \{v \in V : \mathbf{P}(v = v^\star | \mathcal{O}_{\mathcal{U}}) > 0\}$. Then*

$$\mathcal{T}^c_{\mathcal{U},t} \triangleq \{h \in (-\infty, t] : h = d(v, c) - d(v, u_1) - t_1 \text{ for some } v \in \mathcal{B}_{\mathcal{U}}\} \tag{5.29}$$

is the set of possible infection times of c that are smaller than t.

Algorithm 12 D-ON - Update II - deterministic epidemic

```
Require: $\mathcal{B}$, $\mathcal{O}_{\bar{t}}$, $\omega_1 \triangleq (u_1, t_1) \in \mathcal{O}^+_{\tau^\star}$,
        $(u, t_u)$ new positive observation
  $\mathcal{O}^+_t \leftarrow \mathcal{O}^+_{\bar{t}} \cup (u, t_u)$
  $\mathcal{O}^-_t \leftarrow \mathcal{O}^-_{\bar{t}}$
  for $v \in \mathcal{B}$ do
      if $d(u, v) - d(u_1, v) \neq t_u - t_1$ then
          remove $v$ from $\mathcal{B}$
  for $\omega \triangleq (w, \emptyset) \in \mathcal{O}^-_t$ do
      for $v \in \mathcal{B}$ do
          if $d(w, v) - d(u_1, v) < t - t_1$ then
              remove $v$ from $\mathcal{B}$
```

Again, Definition 5.20 does not depend on the choice of $(u_1, t_1) \in \mathcal{O}_{\mathcal{U}}$. The next proposition extends Proposition 5.14 to online localization.

Proposition 5.21. *Let $\mathcal{U}$ be a set of sensors, $c \in V \backslash \mathcal{U}$, $\mathcal{O}_{\mathcal{U}}, \mathcal{B}_{\mathcal{U}}$ as in and Definition 5.13 and fix $(u_1, t_1) \in \mathcal{O}_{\mathcal{U}}$ arbitrarily. Call t_c the infection time of c and define*

$$\begin{aligned} b_{\mathcal{U}}(c, h) &\triangleq \{v \in \mathcal{B}_{\mathcal{U}} : \mathbf{P}(v = v^\star | t_c = h) > 0\} \\ &= \{v \in \mathcal{B}_{\mathcal{U}} : h = d(v, c) - d(v, u_1) + t_1\}, \end{aligned}$$

$$\begin{aligned} \widetilde{b}_{\mathcal{U}}(c) &\triangleq \{v \in \mathcal{B}_{\mathcal{U}} : \mathbf{P}(v = v^\star | t_c > t) > 0\} \\ &= \{v \in \mathcal{B}_{\mathcal{U}} : t < d(v, c) - d(v, u_1) + t_1\}. \end{aligned}$$

Then at time t, g^{SIZE} can be computed as,

$$g^{\text{SIZE}}_{\mathcal{U}}(c) = \sum_{h \in \mathcal{T}^c_{\mathcal{U},t}} \mathbf{P}(v^\star \in b_{\mathcal{U}}(c, h)) \cdot (|\mathcal{B}_{\mathcal{U}}| - |b_{\mathcal{U}}(c, h)|) + \mathbf{P}(v^\star \in \widetilde{b}_{\mathcal{U}}(c)) \cdot (|\mathcal{B}_{\mathcal{U}}| - |\widetilde{b}_{\mathcal{U}}(c)|). \tag{5.30}$$

Proof. Follows from the definition of $g^{\text{SIZE}}_{\mathcal{U}}$, $\mathcal{T}_c$ and $b_{\mathcal{U}}$. □

For g^{DRS}, let $X_c = 1$ if there exists $v \in \mathcal{B}_{\mathcal{U}}$ such that the infection time t_c of c is larger than t (i.e., such that $d(v, c) - d(v, u_1) - t_1 > t$), $X_c = 0$ otherwise. Then, the value of DRS-Gain at time t is defined as

$$g^{\text{DRS}}_{\mathcal{U}}(c) \triangleq |\mathcal{T}^c_{\mathcal{U},t}| + X_c. \tag{5.31}$$

As in Section 5.5.2, we use the same definition of g^{DRS} for both deterministic and non-deterministic epidemics. Instead, an approximation of g^{SIZE} is given in Section 5.8.5.

5.8.5 Approximate Size-Gain for the Non-deterministic Case

When epidemics spread deterministically, Proposition 5.14 and 5.21 show that, for any candidate sensor c, the probability of it being infected at time h, can be computed summing over the possible the nodes $v = v^\star$ such that c is infected at time h. We limit ourselves to give a generalization of Proposition 5.21 to non-deterministic epidemic for the online localization setting For offline localization, Proposition 5.14 can be generalised to non-deterministic epidemic analogously. We adopt the notations of Section 5.6.

Proposition 5.22. *Let t_c be the infection time of $c \in V \backslash \mathcal{U}$ and t'_c, t''_c the minimum and maximum values for t_c given $\mathcal{O}_t$, then*

$$t'_c \geq \min_{v \in \mathcal{B}} \left(\max_{(u,t_u) \in \mathcal{O}_t, t_u \neq \emptyset} \left\{ d(c,v) - d(u,v) + t_u - \varepsilon(d(c,v) + d(u,v)) \right\} \right),$$

$$t''_c \leq \max_{v \in \mathcal{B}} \left(\min_{(u,t_u) \in \mathcal{O}_t, t_u \neq \emptyset} \left\{ d(c,v) - d(u,v) + t_\omega + \varepsilon(d(c,v) + d(u,v)) \right\} \right)$$

Proof. We prove the bound for t'_c, the one for t''_c is analogous. Take $v \in \mathcal{B}$. If $v = v^\star$, then for every $(u, t_u) \in \mathcal{O}_t$

$$t'_c \geq d(c,v) - d(u,v) + t_u - \varepsilon(d(c,v) + d(u,v)), \tag{5.32}$$

hence

$$t'_c \geq \max_{(u,t_u) \in \mathcal{O}_t} \left\{ d(c,v) - d(u,v) + t_u - \varepsilon(d(c,v) + d(u,v)) \right\}. \tag{5.33}$$

The bound follows then from the fact that $v^\star$ can be any node in $\mathcal{B}$. □

For $h \in [t'_c, t''_c]$, let $a(c,h)$ be the set of nodes v that satisfy (5.4) and (5.26) with $v = v^\star$ for all observations in $\mathcal{O}_t \cup \{(c,h)\}$, and let $\widetilde{a}(c)$ be the set of nodes v that satisfy (5.26) at time t for all observations in $\mathcal{O}_t \cup \{(c,\emptyset)\}$. Then we define

$$g_{\mathcal{U}}^{\text{SIZE}}(c) = \int_{\min(t'_c,t)}^{\min(t''_c,t)} (|\mathcal{B}| - |a(c,h)|) f_{t_c}(h) dh + (|\mathcal{B}| - |\widetilde{a}(c)|)(1 - F_{t_c}(t)), \tag{5.34}$$

where $f_{t_c}(\cdot)$ denotes the density of the infection time t_c of c conditioned on $\mathcal{O}_t$ and F_{t_c} is its cumulative function.

Let $(u_0, t_{u_0}) \in \mathcal{O}_{\mathcal{T}^\star}$ and, for $h \in \mathbb{R}$, let us denote by J_h the interval $[h - \frac{1}{2}, h + \frac{1}{2}]$, by J'_h the interval $[h - \frac{1}{2} - t_{u_0}, h + \frac{1}{2} - t_{u_0}]$. In order to compute (5.34), we make the following approximations:

1. we approximate the integrand with a stepwise constant function with steps of unity length centered around the integer values in $[t'_c, t''_c]$, i.e.

$$\mathrm{E}[g_{\mathcal{U}}^{\mathrm{SIZE}}(c)] \approx \sum_{h\in\mathbb{Z}, h\in[t'_c,t''_c], h\leq t} (|\mathcal{B}|-|a(c,h)|)\mathbf{P}(t_c \in J_h|\mathcal{O}_t)+(|\mathcal{B}|-|\widetilde{a}(c)|)\mathbf{P}(t_c > t|\mathcal{O}_t);$$

2. we compute $\mathbf{P}(t_c \in J_h|\mathcal{O}_t)$ by summing over $\mathcal{B}$:

$$\mathbf{P}(t_c \in J_h|\mathcal{O}_{i-1}) = \sum_{v\in\mathcal{B}} \mathbf{P}(t_c \in J_h|v = v^\star, \mathcal{O}_t)\mathbf{P}(v = v^\star|\mathcal{O}_t).$$

   In order to further limit the computational costs, if $\mathbf{P}(v = v^\star|\mathcal{O}_{i-1}) > 0$, we approximate

$$\mathbf{P}(v = v^\star|\mathcal{O}_t) \approx \frac{\mathbf{P}(v = v^\star)}{\mathbf{P}(v^\star \in \mathcal{B})}, \tag{5.35}$$

   i.e., we ignore the fact that, conditioned on the observations in $\mathcal{O}_t$ the probability of a node being the source can differ from the (rescaled) prior. Moreover, we approximate $\mathbf{P}(t_c \in J_h|\mathcal{O}_t)$ as follows. We take (u_0, t_{u_0}) as reference observation[3] and we approximate $\mathbf{P}(t_c \in J_h|\mathcal{O}_t) \approx \mathbf{P}(t_c - t_{u_0} \in J'_h)$.[4]

An important side-effect of the approximation of $\mathbf{P}(t_c \in J_h)$ is that the event $g_{\mathcal{U}}^{\mathrm{SIZE}}(c) = |\mathcal{B}|$, i.e., no node is a valid candidate source after adding c, might have a positive *weight* in the computation of $\mathrm{E}[g_{\mathcal{U}}^{\mathrm{SIZE}}]$. Specifically, there might be a value of h such that $\mathbf{P}(t_c - t_{u_0} \in J'_h) > 0$ but $|a_{c,h}| = 0$. This can lead our algorithm to slow down by choosing sensors that do not reduce the number of candidate sources. We address this problem applying the following heuristic: Whenever the number of candidate sources does not decrease in two consecutive steps we restrict the choice of the new sensor to the set of candidate sources $\mathcal{B}$. In fact, if the infection time of at least one node in $\mathcal{B}$ is already observed, adding a sensor in any other node in $\mathcal{B}$ implies that the cardinality of $\mathcal{B}$ decreases at the next step.

---

[3] In case of a large-diameter network, this choice could be optimized taking as reference the sensor u (static or dynamic) which is closer to the candidate source v; for a small-diameter network this would not yield a substantial improvement.

[4] If the time delays are all uniformly distributed with equal expected values, we can normalize $t_c - t_{u_0}$ to obtain a sum of uniform $U([0,1])$ variables, i.e., an Irwin-Hall random variable and the latter probability can be computed exactly. If time delays are uniformly distributed but with different expected values, the probability $\mathbf{P}(t_c - t_{u_0} \in J'_h)$ is not easily computable Bradley and Gupta [2002], hence we approximate the distribution of $t_c - t_{u_0}$ with a Gaussian distribution with mean and variance equal to the mean and variance of $t_c - t_{u_0}$. The latter Gaussian approximation can be used for generally distributed transmission delays.

5.9 Discussion

In this chapter we presented a very general framework for source localization. Using all the information available at each stage of the localization process, we localize the source by progressively updating a set of candidate sources. Furthermore, a fraction of the available sensors are dynamic, i.e., can be adaptively chosen based on the knowledge about the particular epidemic instance and on the progress of the localization process. We showed that this last feature yields a dramatic reduction in the number of sensors needed to localize the source, with respect to the common setting where all sensors are static, i.e., chosen independently of any epidemic.

Due to its flexibility, the presented framework can be applied in many different contexts, fitting the specific constraints of the setting, e.g., those on the number of sensors and on the time at which they can be deployed.

Several research directions could be investigated using the framework and the formalism introduced in this work. First, a natural and realistic extension could attribute a different cost to static and dynamic sensors or could give the sensors a cost that depends on the time at which they are deployed. Second, in order to further decrease the number of sensors needed, we could approximate the set of candidate sources $\mathcal{B}$ with a smaller set $\overline{\mathcal{B}} \subseteq \mathcal{B}$ excluding the nodes that have a small probability to be the source. Clearly this approximation leads to possible errors (i.e., cases in which the source $v^\star$ is removed from the set of candidates) but, depending on the budget available, a favourable trade-off between cost and precision could be achieved.

A more theoretical and very interesting line of work is the investigation of upper bounds for the number of dynamic sensors needed to reach $\mathcal{B} = \{v^\star\}$ for special classes of networks, e.g., trees.

An interesting and closely related line of work would investigate source localization and sensor placement in adversarial settings, e.g., where the epidemic spread is designed to obfuscate the identity of the source [Fanti et al., 2015] or where an adversary knows the identity of the static sensors and chooses the source in order to maximize the difficulty of source localization. We believe that these settings would require different assumptions about the sensors: for example, sensors that can be iteratively moved in the network or that can reveal information about the infection provenance could be considered.

Conclusion

In this dissertation, we studied the localization of the source of an epidemic in a network. We mostly focused on two questions that have a strong practical impact: *What is the minimum amount of information needed to detect the source? How can we optimize the collection of information for source localization, when we have a constraint on the amount of information we can collect?*

In Chapter 2, we studied the minimum number of observations with which, on a given network, the identification of the source can, under certain hypothesis, be guaranteed. We bounded this quantity in the special case of $\mathcal{G}(N, p)$ random networks. Combining the derived bounds and an experimental analysis, we showed that on $\mathcal{G}(N, p)$ the amount of observations needed follows a non-monotonic behavior as a function of p. This reveals that we can identify parameter regimes in which localizing the source is substantially more difficult than in other regimes.

In Chapters 3 and 4, we turned to optimizing the choice of the nodes to observe, which we called sensors, working in a setting where this choice is made *a priori*, i.e., independently of any epidemic instance.

In Chapter 3, we restricted our attention to trees where the uniqueness of paths makes the problem of sensor placement simpler. For this case, we proposed a polynomial-time dynamic-programming approach that solves the problem in the setting where only K nodes can be chosen as sensors. Our approach can be used to optimize several relevant metrics for source localization.

In Chapter 4, considering general networks, we showed that the optimal sensor set depends not only on the topology of the network, but also on the variance of the node-to-node transmission delays. We considered both a low-variance regime and a high-variance regime for the transmission delays and, in both cases, we proposed algorithms for sensor placement. Furthermore we experimentally showed that, compared to state-of-the-art strategies for sensor placement, our methods have a better performance in terms of source localization accuracy for both the low- and the high-variance regimes.

Finally, in Chapter 5, we proposed a more general framework for source localization where some sensors, called dynamic sensors, can be chosen while the epidemic spreads and the localization process progresses. More specifically, the dynamic sensors can be chosen online, taking advantage of all the already available information about the epidemic. Our experimental analysis shows that, by using dynamic sensors, we dramatically outperform a static strategy that uses the same number of sensors. Furthermore, even with high-variance transmission delays, the use of dynamic sensors makes it possible to localize the source with extremely few sensors, especially in real-world networks.

Further Research Directions

At the end of this journey, several interesting questions remain open. In the final discussion of each chapter we listed some of the most relevant questions, or possibilities for extensions, that arose during our study. We list here a few more general directions that we believe are among the most interesting for future research in the field.

First, in a theoretical direction, it would be interesting to derive possibility / impossibility bounds for source localization and to answer questions such as the following two: *How does the difficulty of localizing the source change while the epidemic progresses*? *Are there diffusion models or networks that make it impossible to guarantee that the source is localized correctly or within a desired precision?* The derivation of punctual statements about the difficulty of source localization is very challenging and requires defining models that, though remaining of practical interest, are amenable to theoretical analysis. Some interesting research in this direction was recently conducted, for example, by Fanti et al. [2015] who studied diffusion protocols to obfuscate the source.

Second, in the direction of developing more scalable approaches, it would be worth studying methods for localizing the source without any knowledge of the entire network or of the full matrix of node-to-node distances. Such methods would rely instead either on a partial network knowledge or, more interestingly, only on extremely local information, e.g., the egonet-topology of each node.

Finally, combining source localization and the development of immunization strategies would be of interest. In fact, it is often the case that the tasks of detecting the source and of containing the epidemic expansion come hand-in-hand. Furthermore, both problems can be expressed within the same framework of an iterative loop of data collection and decision making. Hence, the study of combined approaches that could enhance the performance of both tasks at the same time would be greatly beneficial.

Bibliography

N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. Social fMRI: investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 7(6), 2011. [Cited on pages 71 and 86]

M. Al Qathrady, A. Helmy, and K. Almuzaini. Infection tracing in smart hospitals. In *Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2016. [Cited on page 93]

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), 2014a. [Cited on pages 17 and 19]

F. Altarelli, A. Braunstein, L. Dall'Asta, A. Ingrosso, and R. Zecchina. The patient-zero problem with noisy observations. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(10), 2014b. [Cited on page 19]

N. Antulov-Fantulin, A. Lancic, H. Stefancic, M. Sikic, and T. Smuc. Statistical inference framework for source detection of contagion processes on arbitrary network structures. In *Self-adaptive and Self-organizing Systems Workshops (SASOW)*. IEEE, 2014. [Cited on page 18]

A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286, 1999. [Cited on page 113]

Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihal'ak, and L.S. Ram. Network discovery and verification. *Journal on selected areas in communications*, 24(12), 2006. [Cited on page 27]

N. Berger, C. Borgs, J. T. Chayes, and A. Saberi. On the spread of viruses on the internet. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms*. SIAM, 2005. [Cited on page 2]

J. Berry, W.E. Hart, C.E. Phillips, J.G. Uber, and J. Watson. Sensor placement in municipal water networks with temporal integer programming models. *Journal of Water Resources Planning and Management*, 132(4), 2006. [Cited on pages 23, 77, 88, and 111]

P. Billingsley. *Probability and measure.* John Wiley & Sons, New York, 1995. [Cited on page 80]

B. Bollobás. Almost every graph has reconstruction number three. *Journal of Graph Theory*, 14(1), 1990. [Cited on page 1]

B. Bollobás, D. Mitsche, and P. Pralat. Metric dimension for random graphs. *The Electronic Journal of Combinatorics*, 20(4), 2013. [Cited on pages 26, 29, 31, 32, 34, 39, 43, 44, 45, and 46]

D.M. Bradley and R.C. Gupta. On the distribution of the sum of n non-identically distributed uniform random variables. *Annals of the Institute of Statistical Mathematics*, 54(3), 2002. [Cited on page 136]

D. Brockmann and D. Helbing. The hidden geometry of complex, network-driven contagion phenomena. *Science*, 342(6164), 2013. [Cited on page 19]

J. Cáceres, M.C. Hernando, M. Mora, I.M. Pelayo, M.L. Puertas, C. Seara, and D.R. Wood. On the metric dimension of cartesian products of graphs. *SIAM Journal of Discrete Mathematics*, 21(2), 2007. [Cited on pages 9, 21, 22, and 70]

E. J. Candès and C. Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6), 2014. [Cited on page 23]

L.E. Celis, F. Pavetić, B. Spinelli, and P. Thiran. Budgeted sensor placement for source localization on trees. In *Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS)*, 2015. [Cited on pages 13, 22, and 57]

US Census. California road network. http://www.census.gov/geography.html. *Last accessed on 2017-10-30.* [Cited on pages 71 and 86]

M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *International World Wide Web conference*, 2009. [Cited on page 72]

G. Chartrand, L. Eroh, M.A. Johnson, and O.R. Oellermann. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, 105(1), 2000. [Cited on pages 21, 27, 28, and 29]

X. Chen, X. Hu, and C. Wang. Approximability of the minimum weighted doubly resolving set problem. In *20th Annual Int. Computing and Combinatorics Conference (COCOON)*, 2014. [Cited on pages viii, 9, 10, 11, 20, 21, 25, 26, 31, 49, 50, 52, 76, 77, 78, and 117]

V. Colizza, A. Barrat, M. Barthélemy, and A. Vespignani. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences of the USA*, 103(7), 2006. [Cited on pages 113 and 114]

S. Dhamal, K.J. Prabuchandran, and Y. Narahari. Information diffusion in social networks in two phases. *Transactions on Network Science and Engineering*, 2016. [Cited on page 22]

W. Dong, W. Zhang, and C. W. Tan. Rooting out the rumor culprit from suspects. In *IEEE International Symposium on Information Theory (ISIT)*, 2013. [Cited on pages 17 and 73]

K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis. An efficient curing policy for epidemics on graphs. *Transactions on Network Science and Engineering*, 2014. [Cited on pages 2 and 22]

A. El Badia and T. Ha-Duong. On an inverse source problem for the heat equation. application to a pollution detection problem. *Journal of inverse and ill-posed problems*, 10(6), 2002. [Cited on page 23]

P. Erdős and A. Rényi. On random graphs. *I. Publ. Math. Debrecen*, 6, 1959. [Cited on pages 26, 32, and 113]

P. Erdős and A. Rényi. On two problems of information theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 8, 1963. [Cited on page 27]

G. C. Fanti, P. Kairouz, S. Oh, and P. Viswanath. Spy vs. spy: rumor source obfuscation. In *SIGMETRICS*, 2015. [Cited on pages 23, 92, 137, and 140]

M. Farajtabar, M. Gomez Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song. Back to the past: source identification in diffusion networks from partially observed cascades. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2015. [Cited on page 21]

L. C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977. [Cited on page 21]

M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 29. W.H. Freeman New York, 2002. [Cited on page 26]

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979. [Cited on page 27]

D. Golovin and A. Krause. Adaptive submodularity: theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42, 2011. [Cited on page 22]

M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *SIGKDD International conference on knowledge discovery and data mining*. ACM, 2010. [Cited on page 1]

M. Gomez Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *International Conference on Machine Learning*, 2011. [Cited on page 1]

H.L. Gray and P.L. Odell. On least favorable density functions. *SlAM Review*, 9, 1967. [Cited on pages 89 and 111]

P. D. Grünwald. *The minimum description length principle.* MIT press, 2007. [Cited on page 18]

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. [Cited on page 22]

F. Harary and R.A. Melter. On the metric dimension of a graph. *Ars Combin.*, 2(1), 1976. [Cited on pages 27 and 28]

M. Hauptmann, R. Schmied, and C. Viehmann. Approximation complexity of metric dimension problem. *Journal of Discrete Algorithms*, 14, 2012. [Cited on pages 27, 50, 51, and 52]

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2), 1983. [Cited on page 54]

S. Janson, T. Luczak, and A. Rucinski. *Random graphs*, volume 45. John Wiley & Sons, 2011. [Cited on page 33]

F. Ji and W. P. Tay. An algorithmic framework for estimating rumor sources with different start times. *Transactions on Signal Processing*, 2017. [Cited on page 17]

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. [Cited on page 16]

Y. Kanoria, A. Montanari, et al. Majority dynamics on trees and the dynamic cavity method. *The Annals of Applied Probability*, 21(5), 2011. [Cited on page 18]

O. Kariv and S.L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal of Applied Mathematics*, 37, 1979. [Cited on pages 88 and 111]

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD International conference on knowledge discovery and data mining.* ACM, 2003. [Cited on pages 2 and 23]

S. Khuller, B. Raghavachari, and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70(3), 1996. [Cited on pages 27 and 28]

J. Kratica, M. Čangalović, and V. Kovačević-Vujčić. Computing minimal doubly resolving sets of graphs. *Computers & Operations Research*, 36(7), 2009. [Cited on pages 9 and 25]

A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6), 2008. [Cited on page 23]

A. Kumar, V. Borkar, and N. Karamchandani. Temporally agnostic rumor source detection. *Transactions on Signal and Information Processing over Networks*, 2017. [Cited on page 20]

J. Leskovec, L. A Adamic, and B. A. Huberman. The dynamics of viral marketing. *Transactions on the Web*, 1(1), 2007a. [Cited on pages 2 and 23]

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD International conference on knowledge discovery and data mining.* ACM, 2007b. [Cited on page 23]

J. Lessler, N. G Reich, R. Brookmeyer, T. M. Perl, K. E. Nelson, and D. A. T. Cummings. Incubation periods of acute respiratory viral infections: a systematic review. *The Lancet infectious diseases*, 9(5), 2009. [Cited on page 72]

X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson. Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements. *Review of Scientific Instruments*, 87(4), 2016. [Cited on pages 8 and 23]

A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Physical Review E*, 90(1), 2014. [Cited on pages 18, 19, and 91]

A. Louni and K.P. Subbalakshmi. A two-stage algorithm to estimate the source of information diffusion in social media networks. In *IEEE INFOCOM Workshop on Dynamic Social Networks*, 2014. [Cited on pages 17, 21, 73, and 88]

A. Louni, A. Santhanakrishnan, and K. P. Subbalakshmi. Identification of source of rumors in social networks with incomplete information. In *ASE International conference on Social Computing (SocialCom)*, 2015. [Cited on pages 17 and 21]

W. Luo and W. P. Tay. Identifying infection sources in large tree networks. In *Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).* IEEE, 2012. [Cited on pages 6 and 17]

W. Luo, W. P. Tay, and M. Leng. How to identify an infection source with limited observations. *IEEE Journal of Selelcted Topics in Signal Processing*, 8(4), 2014. [Cited on pages 16 and 20]

W. Luo, W. P. Tay, and M. Leng. Infection spreading and source identification: A hide and seek game. *IEEE Transactions on Signal Processing*, 64(16), 2016. [Cited on page 23]

K. Masood and F. D. Zaman. Investigation of the initial inverse problem in the heat equation. *Journal of heat transfer*, 126(2), 2004. [Cited on page 23]

J.J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, 2012. [Cited on page 113]

E. Mossel and N. Ross. Shotgun assembly of labeled graphs. *arXiv preprint arXiv:1504.07682*, 2015. [Cited on page 1]

P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. *SIGMETRICS Performance Evaluation Review*, 40(1), 2012. [Cited on page 72]

C. Nowzari, V. M. Preciado, and G. J. Pappas. Analysis and control of epidemics: A survey of spreading processes on complex networks. *IEEE Control Systems*, 36(1), 2016. [Cited on pages 2 and 17]

OpenFlights. Route dataset. http://openflights.org/data.html#route. *Last accessed on 2017-10-30.* [Cited on pages 113 and 114]

T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social networks*, 31(2), 2009. [Cited on pages 71 and 86]

R. Pena, X. Bresson, and P. Vandergheynst. Source localization on graphs via ℓ_1 recovery and spectral graph theory. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP).* IEEE, 2016. [Cited on page 23]

M. Penrose. *Random Geometric Graphs.* Oxford Studies in Probability, 2003. [Cited on pages 54 and 113]

P. Pinto, P. Thiran, and M. Vetterli. Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109, 2012. [Cited on pages 2, 6, 17, 20, 21, 73, 79, 80, 89, and 124]

J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E.Steigerwald. Guards - innovative application of game theory for national airport security. In *International joint conference on artificial intelligence*, volume 22(3), 2011. [Cited on page 91]

B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting culprits in epidemics: How many and which ones? In *International Conference on Data Mining (ICDM).* IEEE, 2012. [Cited on pages 16 and 18]

V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *Conference on Decision and Control (CDC).* IEEE, 2013. [Cited on page 2]

M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *SIGKDD International conference on knowledge discovery and data mining*. ACM, 2002. [Cited on pages 2 and 23]

M. Salathé, L. Bengtsson, T. J. Bodnar, D. D. Brewer, J. S. Brownstein, C. Buckee, E. M. Campbell, C. Cattuto, S. Khandelwal, P. L. Mabry, and A. Vespignani. Digital epidemiology. *PLOS computational biology*, 8(7), 2012. [Cited on page 1]

K. Scaman, A. Kalogeratos, and N. Vayatis. Suppressing epidemics in networks using priority planning. *Transactions on Network Science and Engineering*, 2016. [Cited on pages 2 and 22]

J. W. Seaman, P. S. Odell, and D. M. Young. Maximum variance unimodal distributions. *Statistics & Probability Letters*, 3(5), 1985. [Cited on page 97]

E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*, 2012. [Cited on pages 20, 21, 77, and 88]

B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1), 2012. [Cited on page 22]

D. Shah and T. Zaman. Rumors in a network: who's the culprit? *IEEE Transactions on information theory*, 57, 2011. [Cited on pages 2, 17, 18, and 73]

B. Shanmukha, B. Sooryanarayana, and K.S. Harinath. Metric dimension of wheels. *Far East Journal Appl. Math.*, 8(3), 2002. [Cited on page 28]

P.J. Slater. Leaves of trees. *Congr. Numer.*, 14(37), 1975. [Cited on page 27]

Z.C. Somda, M.I. Meltzer, H.N. Perry, N.E. 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. [Cited on page 2]

B. Spinelli, L.E. Celis, and P. Thiran. Observer placement for source localization: the effect of budgets and transmission variance. In *Allerton Conference on Communication, Control & Computing*, 2016. [Cited on pages 22, 69, and 119]

B. Spinelli, L.E. Celis, and P. Thiran. The effect of transmission variance on observer placement for source-localization. *Applied Network Science*, 2017a. [Cited on pages 15, 22, and 69]

B. Spinelli, L.E. Celis, and P. Thiran. Back to the source: an online approach for sensor placement and source localization. In *Intenational World Wide Web conference*, 2017b. [Cited on pages 22 and 93]

B. Spinelli, L.E. Celis, and P. Thiran. A general framework for sensor placement in source localization. *IEEE Transactions on Network Science and Engineering*, 2018. [Cited on page 93]

S. Sundareisan, J. Vreeken, and B. A. Prakash. Hidden hazards: Finding missing nodes in large graph epidemics. In *International conference on Data Mining (SDM)*. SIAM, 2015. [Cited on page 18]

H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *International conference on information and knowledge management*. ACM, 2012. [Cited on page 2]

S. B. Venkatakrishnan, G. C. Fanti, and P. Viswanath. Dandelion: redesigning the bitcoin network for anonymity. In *SIGMETRICS*, 2017. [Cited on page 23]

E. Vergu, H. Busson, and P. Ezanno. Impact of the infection period distribution on the epidemic spread in a metapopulation model. *PLOS one*, 5(2), 2010. [Cited on page 72]

Z. Wang, W. Dong, W. Zhang, and C. W. Tan. Rooting our rumor sources in online social networks: The value of diversity from multiple observations. *Journal of Selected Topics in Signal Processing*, 9(4), 2015. [Cited on page 18]

L. Yartseva, J. E. Simões, and M. Grossglauser. Assembling a network out of ambiguous patches. In *Allerton Conference on Communication, Control & Computing*, 2016. [Cited on page 1]

Y. Yoshida. Almost linear-time algorithms for adaptive betweenness centrality using hypergraph sketches. In *SIGKDD International conference on knowledge discovery and data mining*. ACM, 2014. [Cited on page 88]

S. Zejnilović, J.P. Gomes, and B. Sinopoli. Network observability and localization of the source of diffusion based on a subset of vertices. In *Allerton Conference on Communication, Control & Computing*, 2013. [Cited on pages 21 and 27]

S. Zejnilović, J. Gomes, and B. Sinopoli. Sequential observer selection for source localization. In *IEEE GlobalSIP*, 2015a. [Cited on page 22]

S. Zejnilović, J. Xavier, J. Gomes, and B. Sinopoli. Selecting observers for source localization via error exponents. In *International Symposium on Information Theory (ISIT)*. IEEE, 2015b. [Cited on pages 21 and 91]

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. [Cited on pages 17, 21, 77, 88, and 91]

Z. Zhang, W. Xu, W. Wu, and D. Z. Du. A novel approach for detecting multiple rumor sources in networks with partial observations. *Journal of Combinatorial Optimization*, 2015. [Cited on pages 6 and 21]

K. Zhu and L. Ying. Information source detection in the SIR model: A sample path based approach. In *Information Theory and Applications Workshop (ITA)*, 2013. [Cited on pages 18 and 19]

K. Zhu and L. Ying. A robust information source estimator with sparse observations. *Computational Social Networks*, 1(1), 2014. [Cited on page 19]

K. Zhu and L. Ying. Information source detection in networks: possibility and impossibility results. In *IEEE INFOCOM*, 2016. [Cited on page 18]

K. Zhu, Z. Chen, and L. Ying. Locating the contagion source in networks with partial timestamps. *Data Mining and Knowledge Discovery*, 2015. [Cited on pages 20 and 124]

K. Zhu, Z. Chen, and L. Ying. Catch'em all: Locating multiple diffusion sources in networks with partial observations. *arXiv preprint arXiv:1611.06963*, 2016. [Cited on page 20]

Brunella Marta Spinelli

bmspinelli@gmail.com

Education

2012 - present	**PhD in computer science and communication systems**, EPFL, Lausanne Probabilistic models of information diffusion on networks, estimation of diffusion features, active-learning for sensor placement optimization
2012	**Master Project**, Technical University of Chalmers, Göteborg, Sweden Statistics for alpha-stable point processes
2010-2012	**Master in Mathematics**, Milan State University, Italy, 110/110 *cum laude*
2007-2010	**Bachelor in Mathematics**, Milan State University, Italy, , 110/110 *cum laude*
2002-2007	**Humanities High School**, Istituto Sacro Cuore, 100/100 *cum laude*

Professional experience

2012 - 2017	**Doctoral assistant**, EPFL, Lausanne ◇ Supervisor of several student projects: information-diffusion modeling, mining epidemic data, entity resolution on networks, visualization of network dynamics ◇ Teaching assistant for several classes of bachelor and master level (stochastic models, random networks, ..)
summer 2015	**Intern**, World Health Organisation, Geneva Development of web applications for the analysis of the time-series of tuberculosis cases and for the prediction of the future disease burden

Technical Skills

Programming	Python (NetworkX, Pandas, Numpy, ..) C/C^{++}
Software	Matlab, R, SAS
Technologies	git, LINUX, web (HTML, flask) distributed computing (Condor HTC)

Languages

Italian	Mother tongue
English	Advanced, C2
French	Advanced, C2
German	Intermediate, B1-B2

Main Publications

◇ B. M. Spinelli, E. Celis and P. Thiran. *The effect of transmission variance on observer placement for source-localization.* Applied Network Science, Springer Open, 2017

◇ B. M. Spinelli, E. Celis and P. Thiran. *Back to the Source: an Online Approach to Sensor Placement and Source Localization.* World Wide Web Conference (WWW), 2017.

◇ B. M. Spinelli, E. Celis and P. Thiran. *Observer Placement for Source Localization: the Effect of Budgets and Transmission Variance.* 54th Annual Allerton Conference on Communication, Control, and Computing, 2016.

◇ B. M. Spinelli, E. Celis, F. Pavetic, and P. Thiran. *Budgeted sensor placement for source localization on trees.* Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS), 2015.