Source detection in graphs

Der Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

am 05.07.2023 vorgelegte Dissertation

von

M.Sc. Tobias Weber

Zusammenfassung

Die *Quellensuche in Graphen* ist die Suche nach dem Ursprungs eines Ausbreitungsphänomens in einem Netzwerk. Die Quelle ist ein Knoten des Graphen, der vor der Suche unbekannt ist. Die Quelle könnte zum Beispiel der Ursprung einer Kontamination in einem Wasserverteilungssystem oder einem Logistiksystem für Lebensmittel sein. Ebenso kann der Ursprung einer Krankheit in einem Beförderungsnetz (Flug-, Straßen- oder Bahnverkehr, usw.) von Interesse sein.

Aufgrund der Relevanz der Anwendungen wurden diese Themen von vielen Forschern aus pratischer Sicht betrachtet. Es fehlt bisher eine abstrahierende und generische mathematische Betrachtung. Die vorliegede Arbeit ist ein erster Schritt in diese Richtung. Dafür wird eine allgemeine und einfache Modellvorstellung basierend auf endlichen Graphen und einer konstanten Ausbreitungsgeschwindigkeit des Ausbreitungsphänomens angenommen.

Aufbauend auf dieser Modellierung wird die Problemstellung abstrakt eingeführt. Insbesondere wird zwischen der *online* und der *offline Quellensuche* unterschieden. Die *online Quellensuche* findet gleichzeitig mit dem Ausbreitungsphänomen statt und es ist möglich während der Suche weitere Daten zu sammeln. Die *offline Quellensuche* findet dagegen zeitlich nach dem Ausbreitungsphänomen statt und alle Daten sind zu Beginn der Suche vorhanden.

Eine weitere wichtige Unterscheidung liegt in der *deterministischen* und der *stochastischen Quellensuche*. Die *deterministische* Quellensuche baut auf exakten Daten auf, während die *stochastische Quellensuche* zufällige Fehler in den Daten zulässt und behandelt. Der deterministische Fall is deutlich einfacher als der stochastische und es kann daher vor der Quellensuche angeben werden, welche Daten benötigt werden, um beliebige Quelle zu finden. Hierbei spielen besonders das Konzept der metrischen Dimension eines Graphen und hier vorgestellte Erweiterungen eine Rolle.

Im stochastische Fall werden mindestens die Daten des deterministischen Falls benötigt und zusätzliche Daten um die zufälligen Fehler durch mitteln zu verringern, sodass noch eine hinreichend gute Schätzung der Quelle erreicht wird. Hier ist es a priori nicht möglich anzugeben, welche Daten benötigt werden. Genauso kann die Quelle nicht mehr exakt gefunden werden sondern nur noch geschätzt werden. Für diese Aufgabe wird die lineare Regression verwendet. Über die Fehleranalyse dieser Schätzer werden Heuristiken eingeführt, die angeben, welche Daten gesammelt werden sollten.

Zur Lösung des *stochatischen online Quellenfindungsproblems* wird ein iterativer Algorithmus vorgeschlagen. Dieser besteht aus dem linearen Regressionsschätzer und der Heuristik zur Datensammlung. In jeder Iteration wird hierbei auf Grundlage der bisherigen Daten eine Schätzung der Quelle vorgenommen. Abhängig von der Qualität dieser Schätzung werden entweder weitere Daten gesammelt und eine nächste Iteration angestoßen oder die Schätzung wird akzeptiert und der Algorithmus beendet.

Da die Sammlung der Daten nur heuristisch erfolgt, können keine theoretischen Garantien bezüglich der Konvergenz eines Algorithmus basierend auf dem Schätzer und der Heursitik gegeben werden. Um die Konvergenz zu beweisen wird eine Einschränkung für die Daten, die die Heursitik aussuchen darf, eingeführt und im Algorithmus genutzt um die Konvergenz zu erzwingen.

Die praktische Leistungsfähigkeit dieses Algorithmus wird anhand von numerischen Simulationen gezeigt. Zum einen wird der Algorithmus genutzt um in einer Simulation die Quelle von Herzrhytmusstörungen zu finden und zum anderen um auf allgemeinen Testgraphen die Quelle eines simulierten Ausbreitungsphänomens zu finden.

Eine neue mathematische Theorie für die Quellensuche in Graphen wird in dieser Arbeit eingführt. Innerhalb der Theorie wird ein Algorithmus entwickelt, der das stochastische online Quellenfindungsproblem löst. Die Konvergenz des Algorithmus wird bewiesen und seine Robustheit in numerischen Simulationen gezeigt.

Summary

Source detection in graphs refers to the search for the origin of a spreading signal in a network. The source is an unknown node in the graph, which could be the origin of contamination in a water supply network, food logistics network, or the location of a disease outbreak in a transportation network (air, road, or rail transport). While many researchers have focused on practical applications of this problem, an abstract and generic mathematical examination is lacking. This work provides a general and simple modeling of the problem based on finite graphs and constant speed of the spreading signal.

The problem is defined based on this modeling, with a distinction made between *offline* and *online source detection. Offline source detection* takes place after the signal has propagated through the network and all data is available, while *online source detection* is conducted during the spread of the signal, and new data may be collected during the search. Another important distinction is between *stochastic* and *deterministic source detection*, where the latter is simpler and allows for determining the necessary data, to find any source, before the search.

In the stochastic case, in contrast to the deterministic case, it is not possible to determine the necessary data, to find any source, a priori. In general additional data is required to reduce random errors by averaging, and the source can only be estimated, not found exactly. Linear regression is used for this task, and the error analysis of this estimator leads to heuristics for collecting necessary data. An iterative algorithm is proposed for *stochastic online source detection problem*, consisting of the linear regression estimator and data collection heuristic.

However, as the data is collected heuristically, there are no theoretical guarantees regarding the convergence of the algorithm. To address this, a feasibility constraint on the data is introduced to enforce convergence. The algorithm's practical performance is demonstrated through numerical simulations on simulated cardiac tachycardia and general test graphs.

Overall, this thesis presents a novel mathematical framework for general source detection in graphs, with a new solution algorithm for the stochastic online problem. The algorithm's convergence is proved, and its robustness is shown in numerical simulations.

Acknowledgments

First of all I would like to thank my Professor Sebastian Sager. Without his supportive guidance I would neither have started nor finished this thesis. He always was open to discuss my ideas, suggesting alternatives, and open my mind to ideas I would have not had on my own. Thank you for seeing in me at the beginning what I hoped to become later.

The research group around Sebastian is characterized by open-minded people who are interested not only in mathematics but in all kinds of topics. My colleagues gave me many fruitful discussions about my thesis and often freed my mind from mathematics when my thoughts were blocked after too much concentration on a research problem. For all the coffee we drank during and the time after work I owe you: Felix, Manuel, Clemens, Sören, Duc, Ben and many others. You are the heart and soul of this group and made working there an inspiring and happy experience.

The other part of the optimization department is the group of Professor Volker Kaibel and Professor Gennadiy Averkov. Volker, Gennady, and their students introduced me to a part of the broad field of discrete optimization and graph theory. My thanks to them and to Matthias, Stefan, Wolfgang, Miriam, Julia, and Yauheniya for teaching me linear programming and much more mathematics. I especially owe a lot to Wolfgang for pointing out my many wrong ideas and beliefs and for the heated discussions about computer science and society. Special thanks to Ines and Susanne.

Without the patience of my family, the effort of writing this thesis would not have been possible. First, my parents and my brother and sister made me the person I am today. Then, my wife and daughters supported me morally by motivating me and practically by creating the free time on weekends and evenings that I needed. This thesis is for them: Lena, Sophie and Anke.

Finally, I would like to say that I have not mentioned all the people who have helped me along the way, the list would be endless. All who helped did so without expecting anything in return, sometimes even unasked or unnoticed.

Thank you all.

0 Contents

1 Introduction

Source detection in graphs is a mathematical problem that involves finding the source of a time-dependent spreading process on a given graph. This problem requires collecting time-dependent data about the spreading process on the graph and using this information to estimate the source. The first step of deciding where to collect information is challenging from a complexity point of view, while the second step of source estimation in a finite graph can be done by enumeration. In *offline source detection*, these two steps are performed after the spreading process, while in *online source detection*, the steps may be repeated iteratively during the search while the spreading process is ongoing.

The first step can be interpreted as an experimental design problem, while the second step is usually referred to as estimation, inference, detection, or localization. Both steps have been studied in various settings, especially in Euclidean spaces.

Optimal experimental design is described in [37, 60]. An early idea to solve the experimental design problem is to minimize the variance of the model prediction [101]. This is called G-optimality and is equivalent to the so called D-optimal criterion [61]. The D-optimal criterion [103] maximizes the determinant of the Fisher information matrix. Another criterion is to minimize the variance of the parameter estimates of the model [35], referred to as A-optimality. Our approach is based on the discrete graph structure and inspired by these methods.

The comparison or correlation of the estimation step is solved with linear regression in the case of stochastic problems. The use of regression is not a new idea for source detection problems posed in the usual Euclidean space [12]. In the Euclidean space the difficulty is the nonconvexity of the problem and convexification is performed. In our graph based setting nonconvexity is not a problem, as enumeration over the finite number of possible source nodes is feasible.

The source is a node on the graph that is unique and special because it initiates the *signal spreading process*. The signal spreads from the source over the graph, propagating to all reachable locations. Given the notion of closeness or neighborhood on the graph, the signal spreads from already reached nodes to nearby or neighboring nodes. Therefore, the distance to the source of a node correlates with the time when the signal reaches the node, which leads to the solution of the problem.

To find the source of a *signal spreading process*, the problem is analyzed to find nodes that reveal the maximum amount of information about the source, and the arrival time of the signal at these nodes is collected. For each possible source node, its distance to the measured nodes is compared with the arrival times at these nodes. The node with the best fit or correlation between distance and time is the source estimate. In the online case, a termination check is performed based on the quality of the estimate.

The *source detection problem* has many important applications, and source detection is essential to understand the process and suppress the source in case of negative effects. In case of positive effects, understanding how to support its spread, how to support the source to initiate it more often, or how to create new sources at other locations is important. With the increasing interconnectedness of the world, more networks are created, and their relevance increases, requiring more research on how to use this data efficiently. The thesis provides tools and theory for this task.

This work presents a new mathematical framework for general source detection in graphs, which includes weighted and directed graphs, as well as deterministic or stochastic measurement information. The framework is applicable in both online and offline settings and is based on linear models with known or unknown parameters. The thesis provides different solution approaches for this problem in various settings, including stochastic and deterministic information, and online or offline scenarios. For the most challenging case of the stochastic online problem, the work proposes a new algorithm that is proven to converge in the limit of infinitely many iterations. The algorithm's practical performance is evaluated through numerical simulations, which demonstrate its robustness and flexibility across a wide range of graphs, including those that are weighted or unweighted, directed or undirected.

The work is structured as follows. In Chapter 2 of the thesis presents source detection applications, showcasing how the solution of special stochastic and differential models results in linear time to distance relations. These relations motivate the linear spreading model assumption, which is central to the thesis and its results.

In Chapter 3, the mathematical framework is described in detail. First, the *source detection problem* in graphs is defined in Section 3.2. Next, the deterministic case of the problem is considered in Section 3.3. In the *deterministic offline case* (Subsection 3.3.1), the solution is connected to the (metric) basis of a graph. This concept is extended to fully match our problem case. Possibilities to calculate graph bases efficiently are proposed in Subsection 3.3.2, and in Subsection 3.3.3, an online decomposition approaches to solve the problem are proposed. The final section of the chapter is about the stochastic problem variant. In Subsection 3.4.1 the offline case is presented, while Subsection 3.4.2 is about the online case and provides a proof for convergence in the limit for the *stochastic online source detection* algorithm.

Parts of the chapter are based on [106], which is a pre-print submitted to Automatica. Especially Section 3.2, Subsection 3.3.1 and Subsection 3.4.2 are from the paper. The problem definition is taken from the paper to have a consistent naming. While I found the problem class, provided algorithms and proofs, Sager and Kaibel provided inputs on how to name and define the concepts, present the theory in the paper, pointed out flaws, and suboptimal definitions.

In Chapter 4 a medical application of the proposed algorithm is presented. The chapter was first published in [107]. Subsection 4.1.1 was added in this thesis as an introduction for readers without medical background. In this application, the source of tachycardia in the heart is identified with our algorithm, which could facilitate medical treatment if applied.

The contributions among the authors Weber, Katus, Sager, and Scholz are distributed as follows:

- The authors Katus and Scholz provided the medical application, data, initial research idea, and initial support.
- I developed the theoretical solution and the implementation of the algorithm as well as the numerical results presented in the paper.
- Sager supervised my research with vision, motivation, and feedback, especially helping to identify promising solution directions and potential problems and flaws in the solution.
- Scholz mainly wrote the main part of the paper, especially the medical parts.
- I wrote the smaller theoretical and numerical parts.

Chapter 5 contains simulation results over a wide variety of different graphs. Sections 5.1–5.5 are also from [106]. In these Sections the algorithm implementation is described and performance regarding iterations and source estimation is presented. While I implemented and executed the numerical simulations and provided the results, all three authors contributed to the presentation of these results in the paper. In Section 5.6 additional results are presented, that are original to this work. There results are presented for algorithm convergence, when relaxing a constraint that is central for enforcing convergence in theory and practice.

Chapter 6 concludes and gives possible future research directions. It highlights the contributions of this thesis and the shortcomings that should be overcome in future research. Also, it closes the thesis by restating the important connection to practical applications.

2 | Relevance of source detection in graphs

In this thesis a specific type of process that occurs on networks is considered. For a comprehensive overview of other network processes and networks in general from a practical standpoint, please refer to [87, 21].

Our abstract setting can accommodate a variety of different applications. If the graph's nodes represent individuals and the edges represent friendships or other human relations, then the spreading process could be the dissemination of a new concept, a rumor, or some other form of information. In this context, a disease may also be considered. Typically, this would be modeled at a higher level, with nodes representing cities or countries and edges representing flights or other transportation methods. Similarly, logistic networks that distribute different types of goods may be modeled. The source would be the location where contamination, pollution, or low-quality goods originate. In the case of water or food networks, sources of contamination must be detected.

Deterministic source detection is intimately related to the metric dimension of a graph. Research on metric dimension involves the localization of fires in buildings and LoRaN stations by the coast guard LoRaN stations [27], as well as the classification of chemical compounds [26, 27] and the spread of information or disease [102].

Source detection of voltage sags in an electrical network [58, 89, 69] may also be regarded as a source detection problem. This is not included as an example since it is usually based on directional measurement data. Nonetheless, there are methods that employ not only directional information but also more sophisticated whole-network voltage models [57, 68]. Simplifications (i.e., linearizations) of these methods will also fit within our framework.

Medical applications are neural source detection in the brain for epilepsy research [51] and the mapping and prediction of focal cardiac arrhythmias [107]. The latter is extensively discussed in Chapter 4.

General linear source localization is described in [76], where the challenge of ill-posed problems is tackled, while parameter estimation is considered in [12]. In [49] the detection of objects in astronomical images is examined.

Continuous problems are not considered, as source detection in graphs is a problem that comprises a finite number of elements. In this chapter, several examples are scrutinized before the theoretical framework is outlined. The reader may choose to skip this sections and proceed to Chapter 3.

2.1 Water network source detection

Water is one of the most important resources for human societies. Its distribution to and collection from all members of society is a challenging task, usually performed via networks of sewers and/or pipes. Due to the importance of these networks, optimizing their operation by controllers is an area of active research. To solve the optimization problems, usually simplified—mainly linear—models are used. Nonlinearity in these kinds of problems is due to overflow when the water levels exceed the maximum capacities of channels. This can be treated by tailored algorithmic solution strategies for optimization problems [55].

Detecting the source of pollution in water networks is a practical and relevant task. Much research in this area is focused on the sensors: which chemical or biological markers have to be tested to distinguish different kinds of pollution sources [8, 98, 15, 88]. Additionally, location-dependent visualization and interpolation schemes are used [32]. Some researchers try to detect the source in space. In [67] mainly linear simplified models are used in an optimization framework. Here, flow conditions are assumed to be known and then used to calculate time delays of pollution concentration over the network. These delays are used in a quadratic optimization problem to calculate pollution injection profiles over time for all nodes. A similar flow model-based approach to the offline problem can be found in [78], while in [36] the online case is considered with the goal to place a minimal number of sensors to identify the source.

Ignoring concentrations and just looking at arrival times of the pollution at the sensor locations would simplify the problem from [67] even further and lead directly to our framework. Then the ill-posedness of the problem and the placement of sensors could be treated a bit more rigorously. This is only possible if the pollution starts at a distinct point in time that should be inside the observation time interval. With this approach, one could get information about the location of the source and the starting time of the pollution.

On the other hand, if the pollution is already ongoing, one could simplify the problem by ignoring time and looking at concentration as a measure of distance. Then, one would model the dilution as distance and get information about the source location and amount of pollution material.

These simplifications might seem large. However, the placement of sensors in such networks can be approached with even simpler models [17].

2.2 Acoustic source detection

Humans naturally recognize objects and their positions by hearing their sound. If there is not too much background noise, humans can identify known objects like cars or other people. Most importantly, they can also locate the position of the object. This is possible because humans have two sensors (the ears) with differing positions that give them spatial information about the object. Thus, people can conclude how important it is to consider an object in planning their behavior; usually, the closer the object, the sooner they have to deal with it. Over time, by monitoring the position, they can even track the object's route and speed.

Similarly, one can localize objects that produce sounds by using distributed microphones as a sensor network. Applications range from localizing the talker in a room for camera pointing

[14, 104, 23] to surveillance of outside areas (like crossroads, valleys or industrial facilities) or underwater areas (sonar) [66, 28].

A good overview of algorithms for signal parameter estimation and some historical development (radar and sonar usage in world war II) can be found in [66] while [28] focuses on practical challenges (like wideband signals, near- and far-field, etc.) and the organization of data exchange and network organization.

Acoustic waves traveling through air usually have constant transmission speed, such that the time-distance relationship is linear. The least squares approach in [109] is very similar to our approach but, in contrast, is based on the distances in the Euclidean space. The publication does not only consider acoustic signals but treats general surveillance and source localization for sonar, radar, or radio applications. Sound traveling in water or soil, however, has non-constant, changing speed characteristics. In the area of acoustic source localization, the energy of the signal is also used for source localization [97], resulting in nonlinear relationships. However, the basic dynamics are linear (as shown in [109]) and should therefore also fit into our framework.

The main difference in our approach is that the signals usually travel through Euclidean space, making the use of a graph setting unnecessary. In sonar settings, there is reflection present, which means that the signal may go around obstacles, i.e., the space is not Euclidean (from a shortest path perspective) and might be represented as a graph. Seismic waves were considered in [99, 56], focusing on partial differential equations describing the elastodynamics of the ground. In seismic settings, long-distance signals travel the earth's surface, a manifold close to a sphere. Here, the use of a graph as a discretized representation is appropriate.

2.3 Computer viruses as random models

The spread of computer viruses and fault propagation in information networks has been modeled as spreading phenomena on a graph [96, 29]. In [29] ordinary differential equations are used. This situation is treated in the next section in the context of human diseases. In [96], a stochastic model is used to describe the infection between nodes in the network. Their model and some of their results will be presented first and the connection to our setting will be drawn at the end.

In [96], the network is modeled as an undirected graph $G(V, E)$ on which the virus spreads. An infected node can spread the virus to all its uninfected neighbors. The time until one of them is infected is modeled as an exponential random variable for each edge. All of them are identically distributed with rate λ and independent of each other (without loss of generality they assume $\lambda = 1$). In their setting, the information to locate the unique source node $v \in V$ was just the subgraph of nodes $N \subseteq V$ infected by the virus. It is $v \in N$ because nodes do not recover.

In general graphs, it is difficult to determine the maximum likelihood (ML) estimator of v. Hence, they restrict themselves to regular trees (the graph is infinite, and all nodes have an equal degree) and determine the ML estimator, which they call the rumor center. Then they show that the rumor center is equal to the distance center in the subtree G_N (for any tree) but is different for general graphs. In the end, they show some nice properties of their estimator and an algorithm to calculate it.

Here, the relation of this model to our setting with the constant spreading speed of the virus or signal is of interest.

Proposition 2.3.1. *Given a tree $G(V, E)$ with a source node $v \in V$ and the independent and identically distributed exponential random variables $\tau_{ij}, \forall (i, j) \in E$ to model infection times between neighbors with rate λ, then the arrival time of the virus at any node $n \geq 1$ hops away from the source is distributed according to the Erlang distribution with parameters n and λ, and the expected arrival time is n/λ.*

Proof. As G is a tree, there is a unique path from v to the node of interest with length n. Hence, the random variable is just the sum of n independent and identically distributed exponential random variables.

$$T = \sum_{i=1}^{n} T_i, \ T_i \sim \text{Exp}(\lambda)$$

Therefore, $T \sim \text{Erl}(n, \lambda)$ with corresponding expected value. □

Remark 2.3.2. *The linear relationship between expected arrival time and distance is not restricted to this special distribution. If the spreading time distributions are independent and have the same expected value t_e, then the expected arrival time after at the end of a path with length n is nt_e, due to the linearity of the expected value operator, i.e.,*

$$E[T] = E\left[\sum_{i=1}^{n} T_i\right] = \sum_{i=1}^{n} E[T_i] = nt_e.$$

Remark 2.3.3. *For general graphs, this linear relationship between expected arrival time and distance does not hold anymore. Consider a tree and add one edge to form a loop. Then from any source v, at least one node exists that is reachable by two distinct paths. Let's count the common edges of the two paths as n and the distinct edges of both paths as n_1 and n_2. Then the expected arrival time is*

$$E[T] = \sum_{i=1}^{n} E[T_i] + E\left[\min\left\{\sum_{i=1}^{n_1} T_{i+n}, \sum_{i=1}^{n_2} T_{i+n+n_1}\right\}\right]$$

To calculate the expected value of the random variable

$$T_{min} = \min\left\{\sum_{i=1}^{n_1} T_{i+n}, \sum_{i=1}^{n_2} T_{i+n+n_1}\right\}$$

the following is used

$$1 - F_{T_{min}}(t) = P(T_{min} > t) = P\left(\sum_{i=1}^{n_1} T_{i+n} > t\right) P\left(\sum_{i=1}^{n_2} T_{i+n+n_1} > t\right)$$

which is equal to $(1-F_{n_1}(t))(1-F_{n_2}(t))$ *using shorthand notation for the cumulative distribution functions. The expected arrival time is*

$$E[T_{min}] = \int_0^\infty 1 - F_{T_{min}}(t)dt = \int_0^\infty (1 - F_{n_1}(t))\underbrace{(1 - F_{n_2}(t))}_{\leq 1} dt \leq E[T_{n_1}]$$

and by symmetry, it is $E[T_{min}] \leq \min\left\{E[T_{n_1}], E[T_{n_2}]\right\}$. *Hence, the expected arrival time depends on whether there is a unique path between the source and the destination.*

In general, the signal passes more densely interconnected regions (with more circles) faster than the same (shortest path distance) in a tree. Therefore, the linear relationship between arrival time and distance is lost, but it is still monotone. However, one could compensate for this by manipulating edge weights accordingly.

2.4 Infection spreading as deterministic model

The spread of epidemics can be modeled as a phenomenon on a graph [86, 24, 30, 42, 9]. The spreading can be modeled as a stochastic process as described in the previous section or by deterministic ordinary differential equations (ODEs), which is considered in this section. For ODEs, thresholds that decide if an epidemic dies out fast or spreads over the graph/population, (expected) sizes of infected subpopulations, vaccination schemes to suppress outbreaks, or speed of propagation can be studied. In our context, interest lies in the source detection of epidemics, similar to the approaches using correlation [24], spectrality [38], Bayesian [6], or centrality based estimators [110, 74, 31]. To show the connection between ODEs and our approach, their speed of propagation is considered. Therefore, a simple variant of the standard model first is used.

The so-called SIR-model divides a population into three parts. The first is the susceptible (S) part of the population. These individuals might get infected and become part of the infected (I) subpopulation until they recover and join the recovered (R) part. If there is no recovery, one speaks of the SI-model. In the most simple variant, the population is assumed to be well-mixed, but in this thesis the focus is on the case with a population spread over the nodes of a graph $G(V,E)$, where infection occurs only between individuals at the same node, and the infection is spread between nodes by traveling between nodes along edges.

Definition 2.4.1 (SIR-model on a Graph).

$$\dot{S}_j = -\alpha S_j I_j / N_j + \sum_{n\in\mathcal{N}(j)} (w_{ni}S_n - w_{in}S_j) \qquad \forall\, j \in V \tag{2.1}$$

$$\dot{I}_j = \alpha S_j I_j / N_j - \beta I_j + \sum_{n\in\mathcal{N}(j)} (w_{ni}I_n - w_{in}I_j) \qquad \forall\, j \in V \tag{2.2}$$

$$\dot{R}_j = \beta I_n + \sum_{n\in\mathcal{N}(j)} (w_{ni}R_n - w_{in}R_j) \qquad \forall\, j \in V \tag{2.3}$$

Infections are proportional to the chance that infected individuals (I_j) meet susceptible individuals (S_j) in the total population (N_j) at node j times the rate of infection α. The infected recover with rate β. The individuals at a node j are either susceptible, infected or recovered (i.e., $N_j = S_j + I_j + R_j$). The last term describes the exchange of population between j and its neighbors $\mathcal{N}(i)$, i.e., w_{ij} is the fraction of individuals at j that travel to i per unit of time.

One can simplify the model above by assuming a stable distribution of individuals over nodes (i.e., N_j const.) for example by setting $w_{ij}N_i = w_{ji}N_j$. Then The equation for R_j can be dropped by using relative quantities ($s_j = S_j/N_j$, $i_j = I_j/N_j$). Also, if one is only interested in the initial spread of the epidemics, one can neglect recovery entirely and only consider either s_j or i_j, because both sum to one.

Definition 2.4.2 (Simple SI-model on a Graph)**.**

$$\dot{s}_j = -\alpha s_j(1 - s_j) + \sum_{n \in \mathcal{N}(j)} w_{ni}(s_n - s_j) \qquad \forall\, j \in V \tag{2.4}$$

Without the second term, this is a Bernoulli (more specifically, a logistic) differential equation for every node, which can be solved analytically ([16], quoted after [4]). However, the coupling complicates the solution. If the equation on one node would be influenced by the fixed and known solutions of its neighbors, then the model would correspond to a Riccati equation at each node.

If the graph is an infinite chain the equation can be seen as the discretization of a partial differential equation (PDE) [13]: Fisher's equation [39, 63]. The solution to this PDE is a traveling wave with constant speed. Consequently, also in the discrete graph based setting, in real epidemic data, the linear relationship (when using an appropriate distance measure) between distance to the origin and time of arrival was found and already used to estimate the origin location from arrival time data in different epidemic outbreaks [24].

3 Theory

3.1 Notation

Before the source detection problem is introduced the underlying mathematical structure, concepts and notations are defined.

Our basic structure is a graph. A modern view on graph theory can be found in [33], while the historically interested and German-speaking reader might be interested in the first book on graph theory [64]. A review from a more practical perspective is given in [87].

Definition 3.1.1 (Graph)**.** *The word graph refers to a directed weighted graph $G = (V, E)$ with positive edge lengths $\ell(e) > 0$ for all $e \in E$ and shortest-path-distances $d_{i,j}$ with respect to the length function ℓ from node i to node j for all $i, j \in V$. Let $n := \#V := card(V)$ and $m := \#E := card(E)$.*

The words node and vertex are synonyms.

Definition 3.1.2 (Distances to set)**.** *For a graph $G(V, E)$ and a set $S \subset V$ the vector $d_{i,S}$ is defined as the vector of shortest path distances from i to all nodes in S.*

If the graph G is not clear from the context, the nodes $V(G)$ and the edges $E(G)$ refer directly to the graph G, otherwise just V, E are used.

Definition 3.1.3 (Restricted neighborhood)**.** *Given a directed Graph $G(V, E)$ and weights $w_e, e \in E$ the sets $N^+(v, a) = \{u \in V : \{v, u\} = e \in E, w_e = a\}$ and $N^-(v, a) = \{u \in V : \{u, v\} = e \in E, w_e = a\}$ are neighborhoods of v restricted by edge weight a.*

The vector $\mathbb{1}$ has all entries one.

3.2 Source detection problem

The source detection problem is based on a graph as basic structure (Definition 3.1.1).

Assumption 3.2.1 (Graph)**.** *We assume to have complete knowledge of the graph $G(V, E)$ and the length function ℓ, and hence also of the distance function d.*

In practical applications, the nodes $i \in V$ correspond to spatial locations where measurements are possible. Examples are communities, airports, cities, or countries for infectious diseases, points on a 3d surface grid of the human heart, or sensors in water distribution networks. The edges correspond to connections between the nodes, along which "something may be passed on". This might, e.g., be a viral load via infections, electrical excitation of cells, or transported and diffused pollutant. In the interest of a simplification, and taking the risk that this term does not intuitively match every application, we will simply use the term *signal* to denote this in a general way in the following.

Definition 3.2.2 (Signal Spreading Process)**.** *We consider a dynamic process on a time horizon $\mathcal{T} := [t_s, t_f]$ that originates from an a priori unknown* source $s \in V$ *and spreads the signal via edges to other nodes of the graph. The edge lengths $\ell(e)$ quantify the distances the signal needs to travel to arrive at adjacent nodes.*

Note that the times t_s and t_f are often unknown. The initial time t_s, also called *offset* and indicating when the signal started at source node s, needs to be estimated. The end time t_f is not relevant for the mathematical model. We make some assumptions for the following.

Assumption 3.2.3 (Signal Spreading Process)**.** *We assume that*

1. *The source $s \in V$ is unique.*
2. *Signal spreading takes place in a diffusive way, i.e., a signal is passed on from a node i to all nodes j that are adjacent to i.*
3. *We assume a constant and homogeneous spreading* velocity $1/c > 0$. *Hence, for known distances $d_{s,i}$ we have*
$$t_i := t_s + c \cdot d_{s,i}$$
as the arrival time *at node $i \in V$.*

While the first two assumptions are rather technical, the third assumption is an important restriction of the problem class to a linear model. We note that some applications might need less restrictive assumptions. For example, infections or electric conduction on the heart surface do not have a constant velocity in reality. Also, we are not interested here in measuring the strength of the signal, which may be relevant for certain applications. We now look at the available measurement procedure, abstracted as a data oracle.

Definition 3.2.4 (Data Oracle)**.** *An oracle allows to query nodes $i \in V$ and obtain measurement data r_i. The r_i indicate times t_i when the signal arrived at node i, but with measurement noise,*

$$r_i = t_i + \epsilon_i.$$

Here, $\epsilon_i \in \mathbb{R}$ is a random variable for each $i \in V$. We call the special case of $\epsilon_i = 0 \ \forall \ i \in V$ the deterministic *and the general case the* stochastic version.

Assumption 3.2.5 (Data Oracle Output)**.** *We assume to know the distributions of the measurement errors ϵ_i for all $i \in V$.*

Assumption 3.2.6 (Data Oracle)**.** *We assume that we query the oracle after all relevant times t_i, i.e., data r_i is available at the time of oracle query. In particular, we do not have the possibility to change the process.*

Definition 3.2.7 (Source Detection Problem)**.** *We consider a graph, a signal spreading process, and an oracle as specified in Definitions 3.1.1, 3.2.2, 3.2.4 and the assumptions 3.2.1, 3.2.3, 3.2.5. We denote the task to minimize the number of oracle queries to determine the source node $s \in V$ (possibly up to a tolerance with respect to graph distance) as the* source detection problem.

The queries of the oracle provide (noisy) arrival times r_i, which can be used to infer the unknown offset t_s, the velocity $1/c$, and the source $s \in V$. In this work, we consider the following general approach to source detection.

Definition 3.2.8 (Source Detection)**.** *The general* source detection approach *is: Repeat $i = 1 \dots i_{max}$ rounds of*

S1) choosing k_i nodes $S_i = \{i_1, \dots, i_{k_i}\} \subseteq V$,

S2) querying the oracle to obtain $r_{S_i} \in \mathbb{R}^{k_i}$, and

S3) estimating a current best guess for the source $j^* \in V$.

If $j^* = s$ holds, then we call the approach successful. *The* source detection problem *is to find a successful approach with a minimal number $N = \sum_{i=1}^{i_{max}} k_i$ of oracle queries.*

The special case of $i_{\max} = 1$ is called the *offline version* of the problem. It corresponds to a situation where it is not possible to do calculations between queries to the oracle. The *online version* for $i_{\max} \geq 2$ is not to be confused with more general concepts in online optimization, such as model predictive or dual control. Assumption 3.2.6 relates to the properties of the source detection problem and states that our approach starts after the end of the spreading process at time t_f. Note that some processes such as cardiac excitations have a repetitive nature and a fast timescale, compare [107]. Thus, the results of the considered problem class may find application not only in a posteriori analysis, but also in ongoing processes.

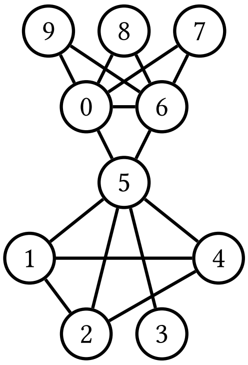
The oracle queries in S2) can be practically difficult and/or expensive, giving rise to our approach to minimize their overall number. Thus, all nodes chosen in S1) have to provide as much information as possible. The problem to identify the corresponding nodes can be seen as an optimal experimental design problem on a graph. The estimation or source inversion problem in S3) can be approached based on regression. Note that the main assumption for this model is a spreading of a signal from the source s to all other nodes via shortest paths at a constant velocity $1/c > 0$. We also assume that the answer of the oracle does not depend on the round in which it is queried. According to the classification in [52], the above setting corresponds to *sensor observations* in contrast to the the *snapshot* or *full information* cases. In the interest of simplicity and if not stated otherwise, we will use notation, definitions, and assumptions from this section, without explicit reference. We shall use the following example for illustration throughout this thesis.

Example 1 (Graph)**.** The graph $G = (V, E)$ has nodes

$$V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

and weights $\ell(e) = 1$ for all undirected edges e in

$$\begin{aligned} E = \{&\{0,5\},\{0,6\},\{0,7\},\{0,8\},\{0,9\},\{1,2\},\{1,4\},\{1,5\},\\ &\{2,4\},\{2,5\},\{3,5\},\{4,5\},\{5,6\},\{6,7\},\{6,8\},\{6,9\}\}. \end{aligned}$$



$i \backslash j$	0	1	2	3	4	5	6	7	8	9
0	0	2	2	2	2	1	1	1	1	1
1	2	0	1	2	1	1	2	3	3	3
2	2	1	0	2	1	1	2	3	3	3
3	2	2	2	0	2	1	2	3	3	3
4	2	1	1	2	0	1	2	3	3	3
5	1	1	1	1	1	0	1	2	2	2
6	1	2	2	2	2	1	0	1	1	1
7	1	3	3	3	3	2	1	0	2	2
8	1	3	3	3	3	2	1	2	0	2
9	1	3	3	3	3	2	1	2	2	0

Figure 3.1: Left: visualization of the example graph. Right: symmetric matrix with shortest path distances $d_{i,j}$.

3.3 Deterministic source detection

In this section the *deterministic version* of the source detection problem (Definition 3.2.8 is discussed, i.e., $\epsilon_i = 0 \ \forall \ i \in V$). This problem class deserves special attention, because it is interesting in its own right. It is the idealized limit case of stochastic versions and algorithmic ideas for this case can be iteratively used in more complex settings. Of practical relevance is the possibility to verify a source via querying the oracle. For the *deterministic version* one possibility is a local enumeration.

Definition 3.3.1 (Source Certificate)**.** *A node $s \in V$ is the* source *of the spreading process if and only if t_s is finite and $t_s < t_j$ for all nodes j with (v_j, v_s) or $(v_s, v_j) \in E$.*

3.3.1 Offline source detection and graph basis

In this subsection we propose a solution to the *deterministic offline version* of the source detection problem, i.e., $i_{\max} = 1$ and $\epsilon_i = 0 \ \forall \ i \in V$. Note that it is purely combinatorial, asking for subsets of V for which the oracle answer allows to infer (or resolve) the source.

We start by considering the special case with $t_s = 0$ and $c = 1$, where the oracle returns $r_i = d_{s,i}$ for $i \in V$. For source detection (Definition 3.2.8) we need to choose a minimal cardinality subset of V in S1) for which we question the oracle in S2). The answer shall enable us to calculate the source in S3) no matter which node in V actually is the source. This concept is known in graph theory as the metric dimension of a graph [27, 26, 80, 102] and depends on the basis of a graph. Classically, the metric dimension of a graph is defined for unweighted graphs, i.e., $\ell(e) = 1\ \forall\ e \in E$. We generalize this to weighted graphs (V, E) with weights $\ell(e) > 0$ for $e \in E$.

Definition 3.3.2 (B-metric Equivalence). *Given a subset $B \subseteq V$, two nodes $i, j \in V$ are* B-metric equivalent *if $d_{i,k} = d_{j,k}\ \forall\ k \in B$.*

Definition 3.3.3 (Metric-Resolving Set). *A set $B \subseteq V$ is* metric resolving, *if $i, j \in V$ are B-metric equivalent if and only if $i = j$.*

Thus, B is metric-resolving if it uniquely defines all $v \in V$ by their shortest path distances to the elements of B.

Definition 3.3.4 (Metric Basis). *A* (metric) basis *B is a metric-resolving set with minimal cardinality.*

If the graph G of the basis is not clear from the context, we use notation $B(G)$.

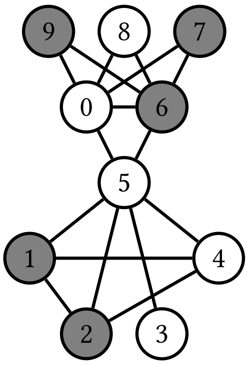
Definition 3.3.5 (Metric Dimension). *Given a weighted graph $G = (V, E)$,* the metric dimension *is the cardinality of one of its metric bases.*

There are different ways to check if a set B is metric-resolving. Equivalently to Definition 3.3.3, one can check if either $\sum_{k \in B} |d_{j,k} - d_{i,k}|$ or (anticipating the stochastic regression case) if $\sum_{k \in B} (d_{j,k} - d_{i,k})^2$ is zero for all pairs of nodes $i, j \in V$ with $i \neq j$. If the value is strictly positive for (the minimum of) all pairs, then B is metric-resolving.

Example 2. The graph from Example 1 has metric dimension 5 and one metric basis is $B := \{1, 2, 6, 7, 9\}$. Figure 3.2 shows that B is a resolving set, as there are no zeros on the off-diagonal. One can show (e.g., by enumeration) that no basis with fewer nodes exists.

Deciding whether a graph has metric dimension less than a given value is NP-complete [59]. Hence, determining the metric dimension even of an unweighted graph is difficult [46]. This computational complexity refers to step S1, the experimental design problem. To find a basis one can enumerate all possible node sets from small to large cardinality until a basis is found. In [46] an $(1 + (1 + o(1)) \log(n))$-approximation algorithms is given which runs in $O(n^3)$, with $n = \text{card}(V)$. The metric dimension can not be approximated within $o(\log(n))$ [45]. If a basis has been found in S1) and the oracle queries returned r_k for all $k \in B$ in S2), the source $s \in V$ can be uniquely determined in S3) by calculating $d_{i,k}$ for all $i \in V$ and $k \in B$ and comparing it to $r_k = d_{s,k}$.

Example 3. Assume that for the basis B from Example 2 the oracle returns $r_{\{1,2,6,7,9\}} = (3, 3, 1, 2, 2)$. Comparison with the full distance table on the right hand side of Figure 3.1 reveals the source node $s = 8$.



j i	0	1	2	3	4	5	6	7	8	9
0	0	2	5	1	4	17	24	16	24	14
1	2	0	5	1	4	17	24	16	24	14
2	5	5	0	2	5	12	13	5	13	9
3	1	1	2	0	3	14	19	11	19	11
4	4	4	5	3	0	5	12	8	12	4
5	17	17	12	14	5	0	5	5	5	1
6	24	24	13	19	12	5	0	4	8	4
7	16	16	5	11	8	5	4	0	4	4
8	24	24	13	19	12	5	8	4	0	4
9	14	14	9	11	4	1	4	4	4	0

Figure 3.2: Left: the graph from Example 1 with the basis in gray. Right: symmetric matrix with entries $\sum_{k\in B}(d_{j,k} - d_{i,k})^2$ for the metric basis from Example 2.

We are interested in a generalization of this concept to arbitrary and a priori unknown velocity $1/c > 0$ and offset $t_s \in \mathbb{R}$. Again, we want to be able to uniquely determine the source, now for arbitrary $c > 0$, t_s, and $s \in V$. While the concepts of a metric basis and of doubly resolving sets [25, 65] can be found in the literature, the spread basis is a novel concept.

Definition 3.3.6 (B-spread Equivalence). *For $B \subseteq V$, two nodes $i, j \in V$ are B-*spread equivalent *if*

$$\exists t_i, t_j \in \mathbb{R}, c_i, c_j > 0 : \quad t_i + c_i d_{i,k} = t_j + c_j\ d_{j,k} \qquad k \in B.$$

Note that with the choice of $t = (t_j - t_i)/c_i$ and $c = c_j/c_i$ this is equivalent to

$$\exists t \in \mathbb{R}, c > 0 : \quad d_{i,k} = t + c\ d_{j,k} \quad \forall\ k \in B.$$

Definition 3.3.7 (Spread-Resolving Set). *A set $B \subseteq V$ is* spread-resolving, *if $i, j \in V$ are B-spread equivalent if and only if $i = j$.*

Definition 3.3.8 (Spread Basis). *A* spread basis *B is a spread-resolving set with minimal cardinality.*

Definition 3.3.9 (Spread Dimension). *Given a weighted graph $G = (V, E)$,* the spread dimension *is the cardinality of one of its spread bases.*

Although the interpretation of a velocity $1/c$ is not well posed for $c = 0$, we will require $c \geq 0$ instead of $c > 0$ in the following minimization problems to avoid open sets. To find a spread basis we consider the objective function

$$J_j(t, c, r_S) = \sum_{k\in S}(t + c\, d_{j,k} - r_k)^2. \tag{3.1}$$

Minimizing this objective with $r_k = d_{i,k}$ and constraint $c \geq 0$ results in an optimal objective value $\phi_{i,j}(S)$ depending on i, j, and S. As above, an equivalent criterion to check if a set $S \subset V$

is spread-resolving is to check if

$$\phi^*(S) = \min_{i,j\neq i\in V} \phi_{i,j}(S) = \min_{i,j\neq i\in V} \min_{t,c\geq 0} J_j(t,c,d_{i,S}) \tag{3.2}$$

is strictly positive, compare Example 4.

Proposition 3.3.10 (Sign symmetric objective)**.** *For any subset S of V the objective values $\phi_{i,j}(S)$ are sign symmetric, i.e., for $i, j \in V$ we have*

$$\begin{aligned} \phi_{i,j}(S) = 0 &\iff \phi_{j,i}(S) = 0 \\ \phi_{i,j}(S) > 0 &\iff \phi_{j,i}(S) > 0. \end{aligned} \tag{3.3}$$

Proof. For $\phi_{i,j}(S) = 0$ with $c > 0$ and t_s we have by equations (3.1) and (3.2):

$$c\, d_{j,k} + t_s = d_{i,k} \,\forall\, k \in S.$$

Reformulating this results in

$$1/c\, d_{i,k} - t_s/c = d_{j,k} \,\forall\, k \in S$$

with new slope $1/c > 0$ and offset $-t_s/c$. Then by equations (3.1) and (3.2) again we have $\phi_{j,i}(S) = 0$. The second part follows from the first due to $\phi_{j,i}(S) \geq 0$. □

Example 4. For the graph from Example 1 the metric basis $B = \{1, 2, 6, 7, 9\}$ is *not* spread-resolving. E.g., for $t_s = c = 1$ we have $d_{8,k} = t_s + d_{6_k}$. The graph has spread dimension 7 and $B^{\text{sp}} := \{1, 2, 3, 6, 7, 8, 9\}$ is a spread basis. Figure 3.3 shows that B^{sp} is spread-resolving, as only diagonal values are zero, and that B is not spread-resolving, as $\phi_{6,8}(B) = \phi_{8,6}(B) = 0$.

$i \setminus j$	0	1	2	3	4	5	6	7	8	9
0	0.00	8.00	8.00	6.86	4.86	1.71	0.75	2.75	2.75	2.75
1	1.71	0.00	1.88	5.73	0.36	0.59	3.43	8.00	8.00	8.00
2	1.71	1.88	0.00	5.73	0.36	0.59	3.43	8.00	8.00	8.00
3	1.71	6.69	6.69	0.00	3.67	0.75	3.43	8.00	8.00	8.00
4	1.71	0.59	0.59	5.18	0.00	0.35	3.43	8.00	8.00	8.00
5	1.71	2.75	2.75	3.00	1.00	0.00	3.43	8.00	8.00	8.00
6	0.37	8.00	8.00	6.86	4.86	1.71	0.00	3.33	3.33	3.33
7	0.59	8.00	8.00	6.86	4.86	1.71	1.43	0.00	6.00	6.00
8	0.59	8.00	8.00	6.86	4.86	1.71	1.43	6.00	0.00	6.00
9	0.59	8.00	8.00	6.86	4.86	1.71	1.43	6.00	6.00	0.00

$i \setminus j$	0	1	2	3	4	5	6	7	8	9
0	0.00	6.80	6.80	1.20	4.00	1.20	0.67	2.00	0.67	2.00
1	1.20	0.00	1.85	0.35	0.32	0.35	2.80	6.80	2.80	6.80
2	1.20	1.85	0.00	0.35	0.32	0.35	2.80	6.80	2.80	6.80
3	1.20	2.00	2.00	0.00	0.67	0.00	2.80	6.80	2.80	6.80
4	1.20	0.55	0.55	0.20	0.00	0.20	2.80	6.80	2.80	6.80
5	1.20	2.00	2.00	0.00	0.67	0.00	2.80	6.80	2.80	6.80
6	0.29	6.80	6.80	1.20	4.00	1.20	0.00	3.14	0.00	3.14
7	0.35	6.80	6.80	1.20	4.00	1.20	1.29	0.00	1.29	5.65
8	0.29	6.80	6.80	1.20	4.00	1.20	0.00	3.14	0.00	3.14
9	0.35	6.80	6.80	1.20	4.00	1.20	1.29	5.65	1.29	0.00

Figure 3.3: Left: matrix with objective values $\phi_{i,j}(B^{\text{sp}})$ for the spread-basis from Example 4. Right: matrix with objective values $\phi_{i,j}(B)$ for the metric basis from Example 2. Both are not symmetric, as $\phi_{i,j}$ can differ from $\phi_{j,i}$.

Our considerations suggest a (not necessarily efficient) approach to find a spread basis. In (3.2), one can detect infeasibility or solve the inner minimization problem analytically and enumerate the outer minimization problem over all (modulo symmetry because of Proposition 3.3.10) pairs of nodes and all subsets S of V. Checking if $\phi^*(S) > 0$ allows to find a spread-resolving set S of minimal cardinality, similar to the metric case.

To close the subsection, we collect some results on bounds for the spread dimension. We are interested in behavior for large $n = \#V$, hence we assume $n \geq 4$ to avoid the discussion of special cases for the following results. If all edges have equal length, a trivial upper bound on the spread dimension is $n - 1$. This bound is active for the special case of complete graphs.

Proposition 3.3.11 (Dimension of Complete Graphs). *Let G be a complete graph with equal weight $\ell > 0$ on all edges. Then the spread dimension of G is $n - 1$.*

Proof. We have $r_s = t_s$ and the same oracle answer $r_i = t_s + c\ell > t_s$ for all $i \in V \backslash \{s\}$ and for all choices t_s and $c > 0$. Hence, the source s can only be identified if either $s \in B$, or if s is the only node in $V \backslash B$. As the spread basis needs to identify all possible s, we have necessarily $\operatorname{card}(B) = n - 1$. □

If the edge weights are not identical, we may even need all n nodes in the spread basis. Thus, complete graphs are the worst case in terms of an upper bound for the spread dimension. However, also other topologies, such as star graphs, may have large spread dimensions.

By definition, the metric dimension is a lower bound for the spread dimension. Furthermore, we have the following lower bound for all graphs.

Proposition 3.3.12 (Lower Bound for Dimension). *Let G be a graph as in Definition 3.1.1 with $n \geq 4$. Then the spread dimension of G is at least 3.*

Proof. Assume a spread basis $B = \{i, j\}$ of cardinality two. Choose $v, w \in V \backslash B$ with $v \neq w$.

If $d_{v,i} = d_{v,j}$ and $d_{w,i} = d_{w,j}$ hold then with

$$c := \frac{d_{v,i}}{d_{w,i}} = \frac{d_{v,j}}{d_{w,j}}$$

we obtain $d_{v,i} = cd_{w,i}$ and $d_{v,j} = cd_{w,j}$, contradicting B being spread-resolving.

Let hence w.l.o.g. $v \in V \backslash B$ be such that $d_{v,i} > d_{v,j}$. Now we can choose $t_s = -d_{v,j}c$ and $c = \frac{d_{j,i}}{d_{v,i} - d_{v,j}} > 0$ and obtain for $v, j \in V$

$$d_{j,k} = d_{v,k}c + t_s = (d_{v,k} - d_{v,j}) \frac{d_{j,i}}{d_{v,i} - d_{v,j}} \quad \forall\, k \in B,$$

contradicting Definition 3.3.7 of a spread-resolving set. □

Again, there are graphs for which this bound is sharp, independent of n.

Proposition 3.3.13 (Lower Bound 3 is Active). *Let $V = \{1, \ldots, n\}$ and $E = \{\{1, 2\}, \{2, 3\}, \ldots, \{n-1, n\}\}$ for $n \geq 4$. The chain graph has spread dimension 3 and $B = \{1, 2, n\}$ is a basis.*

Proof. Let $B = \{1, 2, n\}$. First we note that the distance between two nodes $i, j \in V$ is $d_{i,j} = |i-j|$.

Let w.l.o.g. $a < b \in V$ and $\Delta = b - a > 0$. We consider the three equations

$$d_{a,k} = t_s + c\ d_{b,k} \quad \forall\ k \in B$$

from Definition 3.3.6 and show that no $t_s, c > 0$ exist which satisfy all of them.

If $a > 1$, we have the distances of a to the basis as $d_{a,1} = a - 1$, $d_{a,2} = a - 2$, $d_{a,n} = n - a$ and distances of b accordingly $d_{b,1} = a - 1 + \Delta$, $d_{b,2} = a - 2 + \Delta$, $d_{b,n} = n - a - \Delta$. While the first two equations result in $t_s = -\Delta$ and $c = 1$, the equation for $k = n$ is incorrect with these values.

If $a = 1$, we have distances $0, 1, n - 1$ and $b - 1, b - 2, n - b$, respectively. Here the first two equations result in $t_s = b - 1$ and $c = -1$, but negative c values are not permitted.

Thus, B is a spread-resolving set. With Proposition 3.3.12, it is also an spread basis. □

Summarizing, the spread dimension can be anything between 3 and n for graphs with n nodes. The examples of chain and star graphs show that it is not the absolute number of edges, but rather the graph topology that impacts the spread dimension. Tailored results for specific graph topologies are interesting, but beyond the scope of this work.

3.3.2 Efficient basis calculation

In most of this subsection the word *basis* is used, not distinguishing between a metric basis (Definition 3.3.4) and a spread basis (Definition 3.3.8). In this case the results hold for both bases. If this is not the case, explicitly the word *metric basis* is used. However, with some technicalities, this results should be transferable.

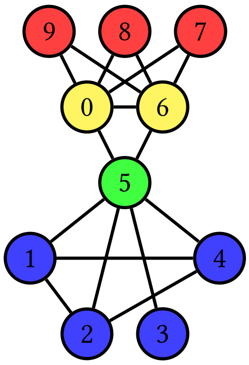
Calculating a basis, i.e., solving the *deterministic offline version* of the source detection problem, is difficult (see Section 3.3.1). Graph decompositions approaches are a tool, to speed up computations (i.e., reducing their time complexity).

One tool to decompose graphs are modules. Modules generalize connected components of graphs, which obviously decompose the metric dimension problem. There exists a recursive way to decompose a graph into modules, which represents all its modules, the modular decomposition [41]. Modular decomposition has many applications in decomposing algorithmic graph problems into smaller subproblems [83, 84]. To argue later that using modular decomposition as an algorithmic tool to speed up computations is viable, it is important that it can be computed in linear time [44, 79].

The definition of modules is extended to weighted (directed) graphs based on the Definition 3.1.3 of a restricted neighborhood.

Definition 3.3.14 (Module)**.** *Given a directed Graph $G(V, E)$ and weights $w_e, e \in E$ a set $M \subseteq V$ is called a* module *if $\forall v \in V \setminus M$ either $M \subseteq N^+(v, a)$ or $M \cap N^+(v, a) = \emptyset$ and $M \subseteq N^-(v, b)$ or $M \cap N^-(v, b) = \emptyset$.*

A module in general is a set where all elements have equal relationship to their neighbors outside of the module. The following example visualizes the concept and also shows a modular partition, which will be used in Subsection 3.3.3.



j i	0	1	2	3	4	5	6	7	8	9
0	0	2	2	2	2	1	2	1	1	1
1	2	0	1	2	2	1	2	3	3	3
2	2	1	0	2	1	1	2	3	3	3
3	2	2	2	0	2	1	2	3	3	3
4	2	1	1	2	0	1	2	3	3	3
5	1	1	1	1	1	0	1	2	2	2
6	2	2	2	2	2	1	0	1	1	1
7	1	3	3	3	3	2	1	0	2	2
8	1	3	3	3	3	2	1	2	0	2
9	1	3	3	3	3	2	1	2	2	0

Figure 3.4: Left: visualization of a modular partition with four members (red, yellow, green, blue). Right: symmetric matrix with shortest path distances $d_{i,j}$, showing in each column to which module it belongs, the light color coding showing the equal shortest paths distances of all module members to all nodes outside of the module.

Example 5 (Modular partition)**.** Consider the graph from Example 1. Some modules in this graph are the sets $\{1\}$, $\{1, 2, 4\}$, and $\{0, 6\}$. Figure 3.4 shows a modular partition of the graph.

Proposition 3.3.15 (Shortest paths distance module)**.** *Given a Graph $G(V, E)$ and a module M a node outside of the module has the same distance to all vertices in the module, i.e., $\forall v \in V \setminus M$ $d_{v,i} = d_{v,j}$ and $d_{i,v} = d_{j,v} \forall\, i, j \in M$.*

Proof: If there is a shortest path from v to any vertex in the module it is also a shortest path to all other vertices in the module (exchanging the last edge into the module). Hence they all have the same length. With the same argumentation the paths in the other direction are also of equal length. □

In this sense the distance between a module and a vertex outside of the module is well defined, especially the distance between disjoint modules.

Definition 3.3.16 (Subresolving)**.** *Given a graph $G(V, E)$ a set $V' \subseteq V$ is* subresolving V'*, if $i, j \in V'$ are equivalent if and only if $i = j$.*

Definition 3.3.17 (Subbasis)**.** *Given a graph $G(V, E)$ a set $V' \subseteq V$ is a* subbasis $B_{V'}$*, if it is a minimal cardinality set subresolving V'.*

In general $B_{V'} \not\subseteq V'$.

Proposition 3.3.18 (Subbasis of modules)**.** *A Subbasis of a module M is contained in the module, i.e., $B_M \subseteq M$.*

Proof. A Subbasis would not be minimal if one would add vertices from outside the module that do not contribute to resolving vertices of the module, as by Proposition 3.3.15 they have the same distance to all vertices in the module. □

Proposition 3.3.19 (Subbasis of complement of modules)**.** *A Subbasis of the complement of a module M contains at most one vertex from the module, i.e.,* $|B_{V\setminus M} \cap M| \leq 1$.

Proof. Because all vertices outside of the module have the same shortest path distance to vertices in the module (Proposition 3.3.15), by the minimality of the basis, only one vertex of the module can be part of B_{SC}. □

Note that exchanging the module node in the subbasis with any other module node results in other subbases, if it exists.

Definition 3.3.20 (Outer path)**.** *Given a graph $G(V, E)$, a set $V' \subset V$ and vertices $v, v' \in V'$ a* outer path *is a path between v, v' including including at least one $o \in V \setminus V'$.*

An outer path might not exist.

Definition 3.3.21 (Shortest outer path)**.** *Given a graph $G(V, E)$, a set $V' \subset V$ and vertices $v, v' \in V'$ the* shortest outer path *is the outer path with minimal distance.*

Definition 3.3.22 (Shortest external outer path)**.** *Given a graph $G(V, E)$, a set $V' \subset V$ and vertices $v, v' \in V'$ the* external shortest outer path *is the part of the outer path that is not in V'.*

If $V' \subset V$ is a module, complement of a module or side of a split, the external shortest outer path is equal for all $v, v' \in V'$ and hence only depends in V'.

Definition 3.3.23 (Extended subgraph)**.** *Given a graph $G(V, E)$ and a module, complement of a module or split side $V' \subset V$ the* extended subgraph $G_{V'}$ *is the subgraph induced by V' extended by V' external shortest outer path. All vertices in V' with connection to the start or end of the external shortest outer path, keep their connection to this vertices as in the original graph.*

If no outer path exists, the extended subgraph is equal to the subgraph induced by V'.

Remark 3.3.24 (Outer path shortening)**.** *If the shortest outer path has more than two vertices outside of V' one would collapse it into only two vertices and one edge, adjusting the length of the middle edge such that the total path length stays the same.*

Proposition 3.3.25 (Extended module subgraph shortest path distances)**.** *Given a graph $G(V, E)$ and a module M all shortest path distances between vertices in M are the same in G and G_M.*

Proof. All vertices in the module are connected to the second and second last vertex of the outer shortest path. All shortest paths between vertices in the module are either this outer path or shorter and inside of the module. This is true for the original graph and the extended module subgraph. Hence, all this (shortest) paths are equivalent in length. □

Proposition 3.3.26 (Extended complement of module subgraph shortest path distances)**.** *Given a graph $G(V, E)$ and a module M all shortest path distances between vertices in $V \setminus M$ are the same in G and $G_{V\setminus M}$.*

Proof. The shortest outer path of $V \setminus M$ just contains one (any) vertex $m \in M$. All module neighbors are connected to m (and any other member of M). Shortest paths between vertices in $V \setminus M$ are either fully contained in $V \setminus M$ or pass through a module neighbor, then (any) module vertex and a module neighbor again. This is true for the original graph and the extended complement of module subgraph. Hence, all this (shortest) paths are equivalent in length. □

Definition 3.3.27 (Constrained subbasis). *Given a graph $G(V,E)$, a set $V' \subseteq V$ and a vertex $v \in V$ a* constrained subbasis *$B_{V'}|v$ is a minimal cardinality set subresolving V' and including v.*

Definition 3.3.28 (Cross resolving subbases). *Given a graph $G(V,E)$ and two sets $V_1, V_2 \subseteq V$ subbases B_{V_1} and B_{V_2} are* cross resolving *V_1, V_2 if no vertex pair $v_1 \in V_1, v_2 \in V_2$ is equivalent with respect to $B_{V_1} \cup B_{V_2}$.*

The basis can now be constructed from the extended subgraphs of one of its modules and the complement of the module.

Lemma 3.3.29 (Modular basis construction). *Given a graph $G(V,E)$ and a nontrivial module M a basis $B(G)$ is given by $B_M(G_M) \cup (B_{V\setminus M}(G_{V\setminus M})|v) \setminus \{v\}$ if $B_M(G_M)$ and $(B_{V\setminus M}(G_{V\setminus M})|v) \setminus \{v\}$ are cross resolving $M, V \setminus M$ in G. The vertex v is the single extension vertex of $B(G_{V\setminus M})$.*

Proof. First it is shown that $B_M(G_M) \cup (B_{V\setminus M}(G_{V\setminus M})|v \setminus \{v\}$ is a resolving set and then that it is minimal.

By Propositions 3.3.25 and 3.3.26 the shortest path distances in the extended subgraphs are equal to the shortest path distances in G between vertices in M and between vertices in $V \setminus M$. Hence $B_M(G_M)$ is subresolving M in G and $(B_{V\setminus M}(G_{V\setminus M})|v \setminus \{v\}$ in combination with any vertex from $B(G_M)$, which has the same distance properties in G as v in $G_{V\setminus M}$ for all vertices in $V \setminus M$, is subresolving $V \setminus M$. As $B_M(G_M)$ and $(B_{V\setminus M}(G_{V\setminus M})|v) \setminus \{v\}$ are also cross resolving $M, V \setminus M$ in G their union is resolving V.

Lets assume $B(G_M) \cup (B(G_{V\setminus M})|v) \setminus \{v\}$ is not minimal. Then there exists a basis B' with less elements. By Proposition 3.3.18 $B' \cap M$ is resolving M and by Proposition 3.3.19 $B' \cap (V \setminus M) \cup \{v'\}$ is resolving $V \setminus M$ (v' is any element of $B' \cap M$) and by Propositions 3.3.25 and 3.3.26 this also holds in the respective extended subgraphs. Then either $|B' \cap M| < |B_M(G_M)|$, contradicting that $B_M(G_M)$ is a subbasis, or $|(B' \cap (V \setminus M)) \cup \{v'\}| < |B(G_{V\setminus M})|v|$, contradicting that it is a constrained subbasis. □

This result can be used to calculate the basis of a graph in a recursive algorithm given a modular decomposition tree, traversing it from the bottom to the top. Depending on the graph and its partition, this can be much more efficient than the solution on the original graph. The main problem is that the bases one combines have to be cross resolving.

Definition 3.3.30 (All cross resolving subbasis). *Given a graph $G(V,E)$ and a set $V' \subseteq V$ a subbasis $B_{V'}$ is* all cross resolving *if for no set $W \subseteq V \setminus V'$ any vertex pair $v' \in V', w \in W$ is equivalent with respect to $B_{V'} \cup B_W$, with any subbases B_W.*

Proposition 3.3.31 (All cross resolving metric basis of module). *Given a graph $G(V,E)$ and a module M a subbases B_M with more than one element is all cross resolving if no vertex $m \in M$ has equal distance to all elements of B_M.*

Proof. Because a vertices outside of the module has the same shortest path distance to vertices of B_M (by Proposition 3.3.15) and no vertex in the module has the same shortest path distance to all elements of B_M, no vertex outside of M is equivalent to any vertex in M. □

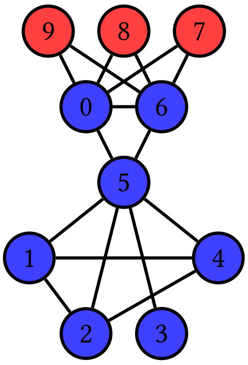
In case not cross resolving subbases come up during tree traversal, expensive recalculation of the basis in that tree node might be necessary. However, depending on the situation one could exploit already calculated information of children of the current node to warm start the calculations.

In the second part splits are considered as a decomposition tool for graphs, here they are defined for weighted and directed graphs.

Definition 3.3.32 (Split)**.** *Given a directed Graph $G(V,E)$ and weights $w_e, e \in E$ a* split *is partition of V in two sets V_1 and V_2 with $|V_1| > 1, |V_2| > 1$ such that there exists sets $W_1^+, W_1^- \subset V_1$ and $W_2^+, W_2^- \subset V_2$ such that $\forall v \in W_1^+, N^-(v,a) \cap V_2 = W_2^-$ and $\forall v \in V_1 \setminus W_1^+, N^-(v,a) \cap V_2 = \emptyset$ and $\forall v \in W_1^-, N^+(v,b) \cap V_2 = W_2^+$ and $\forall v \in V_1 \setminus W_1^-, N^+(v,b) \cap V_2 = \emptyset$. There are no more edges than these between the two sides V_1 and V_2 of the split.*

In a split all edges from V_1 to V_2 have the same weight a and form a complete directed bipartite graph. The same holds true for all the edges in the other direction with a different edge weight b. This is not a strict generalization of our module definition.

Example 6. Consider the graph from Example 1. Some splits in this graph are $(\{1,2,3,4\}, \{0,5,6,7,8,9\})$, $(\{1,2,4\}, \{3,0,5,6,7,8,9\})$ or $(\{1,2,3,4,0,5,6,7\}, \{8,9\})$. Figure 3.5 shows a split of the graph.



j i	0	1	2	3	4	5	6	7	8	9	S
0	0	2	2	2	2	1	2	1	1	1	0
1	2	0	1	2	2	1	2	3	3	3	2
2	2	1	0	2	1	1	2	3	3	3	2
3	2	2	2	0	2	1	2	3	3	3	2
4	2	1	1	2	0	1	2	3	3	3	2
5	1	1	1	1	1	0	1	2	2	2	1
6	2	2	2	2	2	1	0	1	1	1	0
7	1	3	3	3	3	2	1	0	2	2	0
8	1	3	3	3	3	2	1	2	0	2	0
9	1	3	3	3	3	2	1	2	2	0	0
S	0	2	2	2	2	1	0	0	0	0	1

Figure 3.5: Left: visualization of a split (red, blue). Right: symmetric matrix with shortest path distances $d_{i,j}$, showing for each column to which side of the split the vertex belongs (red/blue), including an additional row/column giving the distance of the vertex to the split. The distance of the split is the length of all its edges, i.e., the gap over the split, which might be different in each direction. One can see the decomposition of shortest path lengths across the split.

Proposition 3.3.33 (Shortest paths split decomposition)**.** *Given a Graph $G(V,E)$ and a split V_1, V_2 with $W_1^- \subseteq V_1$ and $W_2^+ \subseteq V_2$ the shortest path distance from $v_1 \in V_1$ to $v_2 \in V_2$ is $d_{v_1,v_2} = d_1 + a + d_2$. Where d_1 is the shortest path distance from v_1 to W_1^-, d_2 is the shortest path distance from W_2^+ to v_2, and a is the length of edges from W_1^- to W_2^+.*

Proof: The two shortest paths from v_1 to W_1^- and from W_2^+ to v_2 with the corresponding edge of length a form a path. There can not exist a shorter path, because then one could construct from it shorter shortest paths for either side of the split from and to v_1, v_2. □

Remark 3.3.34 (Shortest path recursion)**.** *By Proposition 3.3.33 shortest path calculation can be simplified in two ways, one can just calculate all shortest paths from (and to) all vertices to (and from) their side of the split, which already contains all information about every possible shortest path (length). Also given a split decomposition tree one could calculate all shortest paths on much smaller sides of splits traversing the tree from bottom to top.*

Proposition 3.3.35 (Subbasis of split side)**.** *A Subbasis B_{V_1} of one side V_1 of a split V_1, V_2 contains at most one vertex from the other side of the split, i.e., $|B_{V_1} \cap V_2| \leq 1$.*

Proof. If two vertices in V_1 are equivalent with respect to one vertex from V_2 this means that their distance to the split (i.e., to W_1^-) is the same. Therefore by Proposition 3.3.33 their distance to any other vertex in V_2 is the same and they are equivalent to any vertex in V_2. By the minimality of the subbasis at most one vertex of V_2 is in B_{V_1}. □

Proposition 3.3.36 (Extended split side subgraph shortest path distances)**.** *Given a graph $G(V,E)$ and a split V_1, V_2 all shortest path distances between vertices in V_1 are the same in G and G_{V_1}.*

Proof. Shortest paths in G and in G_{V_1} must have the same length, otherwise one would be able to construct from the shorter path a path with equal length in the other graph. If the path is contained in V_1 the construction is trivial. Otherwise one would have to replace the part not in V_1 with a (possibly new and shorter or the same) outer path. □

The metric basis of a graph G can be constructed from the extended subgraphs of the split sides.

Lemma 3.3.37 (Split metric basis construction)**.** *Given a graph $G(V,E)$ and a split V_1, V_2 a basis $B(G)$ is given by $((B_{V_1}(G_{V_1})|v) \setminus \{v\}) \cup (B_{V_2}(G_{V_2})|w) \setminus \{w\}$ if $(B_{V_1}(G_{V_1})|v) \setminus \{v\}$ and $(B_{V_2}(G_{V_2})|w) \setminus \{w\}$ are cross resolving V_1, V_2 in G. The vertex v is the extension vertices of G_{V_1} and w is the extension vertex of G_{V_2}.*

Proof. First it is shown show that $((B_{V_1}(G_{V_1})|v) \setminus \{v\}) \cup (B_{V_2}(G_{V_2})|w) \setminus \{w\}$ is a resolving set and then that it is minimal.

By Proposition 3.3.33 the shortest path distances in the extended subgraphs are equal to the shortest path distances in G between vertices in V_1 and between vertices in V_2. Hence $(B_{V_1}(G_{V_1})|v) \setminus \{v\}$ in combination with any vertex from $(B_{V_2}(G_{V_2})|w) \setminus \{w\}$, which has the same distance properties in G as v in G_{V_1} for all vertices in V_1, is subresolving V_1 in G. The same holds for vertices in V_2 with the mirrored argument (by symmetry). As the two subbases are cross resolving V_1, V_2 in G their union is resolving V.

Lets assume $((B_{V_1}(G_{V_1})|v) \setminus \{v\}) \cup (B_{V_2}(G_{V_2})|w) \setminus \{w\}$ is not minimal. Then there exists a basis B' with less elements. By Proposition 3.3.35 $B' \cap V_1 \cup \{v'\}$ distinguishes elements in V_1 (v' is any element of $B' \cap V_2$) and by Propositions 3.3.36 this also holds in G_{V_1}. The same is true for V_2 (by symmetry). Then either $|B' \cap V_1 \cup \{v'\}| < |B_{V_1}(G_{V_1})|v|$, contradicting that $B_{V_1}(G_{V_1})|v$ is a constrained subbasis, or this contradiction is on the other side of the split (by symmetry). □

Based on this result one can design a recursive algorithm traversing a split decomposition tree from bottom to top, solving the basis problem only on the leaves and combining the solutions on the way up the tree to the root to the full basis. Depending on the graph (and its decomposition) this might be more efficient than the direct solution in the original graph. The problem of not cross resolving bases during algorithm runtime is the same as in the case of modular decomposition.

Proposition 3.3.38 (All cross resolving metric basis of split side). *Given a graph $G(V, E)$ and a split V_1, V_2 a subbases B_{V_1} is all cross resolving if no vertex $v \in V_1$ has distances to all elements in B_{V_1} that are equal to the shortest shortest path distances of W_2^- to all elements in B_{V_1} plus a common positive offset, i.e., $\nexists v \in V_1, w \in W_2^-, \delta > 0$ such that $d_{v,B_{V_1}} = d_{w,B_{V_1}} + \delta$.*

Proof. As all vertices from V_2 have by Proposition 3.3.33 a distance to all vertices in B_{V_1} which is equal to shortest shortest path distances of W_1^+ to all elements in B_{V_1} plus an positive offset, there can not be any vertex in V_2 having the same distances to B_{V_1} as any vertex in V_1. □

For not cross resolving bases special treatment in the algorithm is necessary.

It is beyond the scope of this work to investigate these possibilities to calculate bases in detail. The following example for the metric basis and the modular decomposition tree is given.

Example 7 (Metric basis from modular decomposition tree). Consider the modular decomposition tree of the graph $G(V, E)$ from Example 1 with vertices

$$V_T = \{V, a, b, c, a', \{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{9\}\}$$

representing the following original vertices

$$a = \{1, 2, 3, 4\}$$

$$b = \{0, 6\}$$

$$c = \{7, 8, 9\}$$

$$a' = \{1, 2, 4\}$$

and edges

$$E_T = \{\{V, a\}, \{V, 5\}, \{V, b\}, \{V, c\}, \{a, a'\}, \{a, 3\}, \{b, 0\}, \{b, 6\},$$
$$\{c, 7\}, \{c, 8\}, \{c, 9\}\}.$$

The root of the Tree is the node V, which is the full vertex set of the original graph. The leaves of the tree are single vertex modules that can not be decomposed further. The children

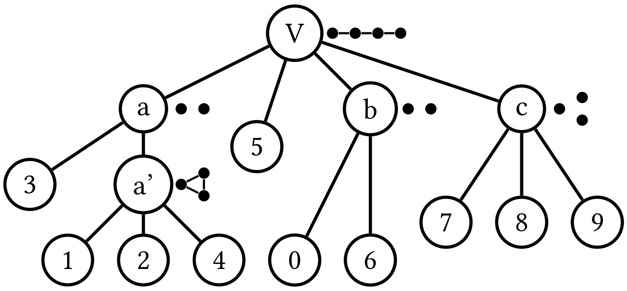


Figure 3.6: Modular decomposition tree of the example graph with quotient graphs depicted right to the non leave nodes of the tree.

of the root are a partition of maximal modules of the whole graph: a, b, c, and 5. The subgraphs induced by the children are further decomposed by their children. The children of a partition a into the modules $\{3\}$ and a'. The children of b are two single vertex sets, as well as the children of c are three single vertex sets.

Then, the tree is traversed bottom-up, in the leaves nothing is to do. For each node the subgraph induced by this module is investigated. The quotient graph of this subgraph defined by the partition given by its children vertices (modules) is formed. For this quotient a basis has to be found, that ideally would include vertices representing lower level modules where already basis elements were selected for solving lower level basis problems.

Looking at a' we have to include two of the vertices $1, 2, 4$ into the basis because its quotient is the complete graph with three vertices. For b we have to take 0 or 6 and for c two from $7, 8, 9$ because they are complement of complete graphs with two and three vertices. As a has as quotient the complement of the complete graph with two vertices ($\{3\}$ and a') and in a' we already choose a basis element, we do chose a' as our basis element here. Finally in the root the quotient is a path graph with four vertices a, $\{5\}$, b, and c. As we already chose three basis elements a, b, and c from the underlying children, we are done.

Hence, a metric basis of our graph must contain: two of the vertices $1, 2, 4$; 0 or 6; and two of the vertices $7, 8, 9$. The metric dimension is 5.

Possible questions are: which graph classes are suited for this kind of method, from the complexity theoretic viewpoint and the practical side. Modular and split decompositions are not yet investigated for weighted graphs, such that the above results can only be applied to unweighted graphs at the moment. Open source implementations even for unweighted/undirected graphs for the above decompositions are rare or nonexistent, especially in their linear time complexity variants. The only implementation known to the author has no linear time complexity [94].

3.3.3 Online source detection and graph decomposition

In this subsection a solution to the *deterministic online version* of the source detection problem is proposed, i.e., $i_{\max} > 1$ and $\epsilon_i = 0 \; \forall \; i \in V$. The tool that was introduced for basis calculation is now used to derive an online algorithm. Modular decomposition is used to show the general idea for the metric case (velocity one and initial time zero), the extension to the split decomposition and/or unknown speed and initial time should be straightforward. The considerations are based on the directed modular decomposition [79], as modular decomposition for weighted (directed) graphs was not yet investigated in the literature.

Let us recall some known facts. The following is an direct consequence of Definition 3.3.14.

Proposition 3.3.39 (Neighbourhood of modules)**.** *Given a graph $G(V,E)$ and two disjoint modules $M_1, M_2 \subset V$ either all edges from all vertices in M_1 to all vertices in M_2 exists, or none.*

In this sense one can speak of an "edge" from one module to another or neighbouring modules. Hence, it is hence possible to view the graph on a level of disjoint modules.

Definition 3.3.40 (Modular partition)**.** *Given a graph $G(V,E)$ a* modular partition *of G is a partition P of V where all elements of P are modules of G.*

The Example 5 shows such a partition and its neighbour relations on the partition level. This leads to the following definition.

Definition 3.3.41 (Quotient graph)**.** *Given a graph $G(V,E)$ and a modular partition P of G. The* quotient graph *G/P is the graph representing the neighbour relations of the modules in P, i.e., each vertex in G/P represents one module in P and each edge in G/P represents a directed connection between the corresponding modules in G.*

The quotient graph represents all edges between modules in P. All other edges are represented by the subgraphs induced by the modules in the partition.

Definition 3.3.42 (Factor)**.** *Given a graph $G(V,E)$ and a modular partition P of G. A* factor *is the subgraph induced by a module $p \in P$.*

A graph is completely represented by its quotient and factors. As each factor can be split again into a quotient and factors, this leads to a recursive decomposition until factors with only one vertex are left. This structure can be represented as a tree.

Definition 3.3.43 (Modular decomposition tree)**.** *Given a graph $G(V,E)$ a* modular decomposition tree *T of G is a rooted tree with the following properties:*

- *The root of the tree corresponds to the the full vertex set V.*
- *Each leave of the tree corresponds to a vertex in V.*
- *Each vertex $t \in T$ in the tree corresponds to a module $M(t)$ including all vertices of leaves of the subtree rooted at this vertex.*
- *The children of a node $t \in T$, represent a modular partition of the subgraph induced by $M(t)$.*

Note that it is not required that the modular decomposition is the unique modular decomposition that implicitly represents all others. For each node t of the modular decomposition tree T, its *quotient* is the quotient graph for the subgraph induced by $M(t)$ and the partition of this subgraph into modules given by the children of t.

After this repetition of the concept of the modular decomposition tree some of its concepts are extended to state the algorithm.

Definition 3.3.44 (Extended quotient graph)**.** *Given a graph $G(V, E)$ and a modular decomposition tree T of G. The* extended quotient graph $G/P^{ex}(t)$ *of a tree vertex $t \in T$ is its quotient graph extended by the shortest outer path (Definition 3.3.21) of the module it represents in G. The extension adds the vertices and edges of the shortest outer path except start and end vertex. All vertices in G/P (modules of G) with connection to the second or second last vertex of the shortest outer path, keep their connection to this vertices as in the original graph.*

To describe the shortest paths of the extended quotient graph, some basis on shortest paths and modules is needed.

Lemma 3.3.45 (Shortest Path in Module)**.** *Let $M \subseteq V$ be a module of a graph $G(V, E)$ and $i, j \in V$ distinct vertices. If the shortest path $P_{short} \subset V$ between i, j intersects M, then:*

$$P_{short} \cap M = \begin{cases} \{i, j\}, \\ P_{short}, \\ \{p\}, p \in P_{short}. \end{cases}$$

Either the shortest path is completely contained in M, or only its start and end are in M, or one of its vertices is in M.

Proof. Lets first consider the case that $\{i, j\} \subseteq M$ and show that only the first two cases can occur. For $|P_{short}| < 4$ this is trivial. Lets assume $|P_{short}| \geq 4$. To proof that either all or none of the intermediate nodes of the path are in M, assume that a path with consecutive intermediate nodes $a, b \in P_{short}$ with $a \notin M$ and $b \in M$ exists (any non conformant path must have such a pair of nodes). Assume first that a is before b in the path (i.e., closer to i). Then a shorter path can be constructed by just connecting a directly to j instead of b (j and b are in M and hence both neighbours of a). In the case that b is before a, a is connected with i. In both cases a shorter path exists, violating our assumption. Proving the first two cases.

Lets now assume that either i or j or none of the two vertices is in M and show that this enforces the third case. For $|P_{short}| < 3$ this is trivial. Lets assume $|P_{short}| \geq 3$ and $P_{short} \cap B > 1$. Either $i \notin (P_{short} \cap M)$, then one can just connect the vertex before the first of the vertices directly to the last of them (because both of them are in M and hence neighbours of the vertex before the first), or $j \notin (P_{short} \cap M)$, then one connects the vertex after the last to the first. In both cases a shorter path exists, hence at most one vertex can be in M in this case. □

This leads to the following properties.

Proposition 3.3.46 (Shortest path equivalence of quotient graph)**.** *Let P be a modular partition of $G(V, E)$ and $i, j \in V$ vertices of different modules of the partition P, i.e., $i \in P_i \in P$, $j \in P_j \in$*

$P, P_i \neq P_j$. Then the shortest path $\mathcal{P}$ from P_i to P_j in the quotient graph induced by the partition can be used to construct a shortest path from i to j in the original graph with equal length. For all modules in the shortest path in the quotient graph one takes any node in the corresponding modules from the original graph except for P_i and P_j where i and j are chosen. The inverse construction is trivial.

Proof. Every module of $\mathcal{P}$ contains at most one vertex from a shortest path from i to j by Lemma 3.3.45. All paths with at most one vertex per module in $\mathcal{P}$ are contained in the quotient graph with the above construction. The path in the quotient graph and the corresponding in the original have the same length. Therefore, a shortest path in the quotient corresponds to a shortest path in the original graph. □

Proposition 3.3.47 (Shortest path equivalence of extended quotient graph)**.** *Given a graph $G(V, E)$, a modular decomposition tree T of G and a extended quotient graph G/P^{ex} of a node in T a shortest paths between vertices in G/P^{ex} is equivalent to the shortest paths between the nodes of the modules in G that are represented by the vertices in G/P^{ex}. From a shortest path in G/P^{ex} a shortest path can be created by choosing for a vertex in G/P^{ex} representing a module in G any vertex from the module and for the extension part of G/P^{ex} choosing the shortest outer path in G used to create the extension.*

Proof. By Lemma 3.3.45 a shortest path in G/P^{ex} is either inside the non extended quotient part of the graph, or inside the extension shortest outer path, at most having start and/or end vertex inside the non extension part. In the first case the equivalence is due to Proposition 3.3.46. In the second case the equivalence is trivial. □

For a node i of a modular decomposition tree or a (extended) quotient graph, $M(i)$ denotes the module that the node represents in the original graph.

The key problem is, that the distance of a node to itself in the (extended) quotient is zero, while in the original graph it might not be zero, i.e., two nodes in the module represented by the node in the quotient do not have zero distance (unless they are the same node).

Definition 3.3.48 (Indirect Extended Quotient B-metric Equivalence)**.** *Given a graph G, one of its extended quotient graphs G/P^{ex} and a subset $B \subseteq V(G/P^{ex})$, two nodes $i, j \in V(G/P^{ex})$ are* indirect extended quotient B-metric equivalent *if either $\exists i', i'' \in M(i) : d_{j,i} = d_{i',i''}$ or $i \notin B$ and either $\exists j', j'' \in M(j) : d_{i,j} = d_{j',j''}$ or $j \notin B$.*

Definition 3.3.49 (Direct Extended Quotient B-metric Equivalence)**.** *Given a graph G, one of its extended quotient graphs G/P^{ex} and a subset $B \subseteq V(G/P^{ex})$, two nodes $i, j \in V(G/P^{ex})$ are* direct extended quotient B-metric equivalent *if $d_{i,k} = d_{j,k} \ \forall \ k \in (B \setminus \{i, j\})$ and if $i \in B$ $d_{i,k} = d_{j,k} \ \forall \ k \in (B \setminus \{i, j\})$.*

Definition 3.3.50 (Extended Quotient Metric-Resolving Set)**.** *A set $B \subseteq V$ is* extended quotient metric resolving, *if $i, j \in V$ are direct and indirect extended quotient B-metric equivalent if and only if $i = j$.*

Definition 3.3.51 (Extended Quotient Metric Basis)**.** *A* extended quotient metric basis *B is a metric-resolving set with minimal cardinality.*

This directly leads to the following property.

Proposition 3.3.52 (Unique module resolvability in quotient)**.** *Given an extended quotient graph G/P^{ex} and a extended quotient metric basis B one questions for all $i \in B$ the oracle to any node $j \in M(i)$ each. Then, the source can be uniquely attributed to a module in the original graph corresponding to a vertex in G/P^{ex}, if the source is in any of the modules represented by nodes in G/P^{ex}.*

Proof. Lets assume that it is not possible to uniquely attribute the source to a module, then $\exists i, j \in V(G/P^{ex})$ are directly and indirectly extended quotient B-metric equivalent, which is not possible as B is extended quotient metric resolving G/P^{ex}. □

As the modular decomposition is recursively defined/calculated by splitting a graph into quotients and factors the *deterministic online source detection* is recursively finding the source factor in the quotients of the current tree node.

Algorithm 1 Deterministic Online Source Detection

Input: $G(V, E)$, $d_{i,j} \forall i, j \in V$, $\mathcal{V}$ and T
Output: Source j^*

```
1: $t \longleftarrow root(T)$
2: while $t \notin leaves(T)$ do                          ▷ End search in a leave of the tree
3:     Calculate $G/P^{ex}(t)$                              ▷ Definition 3.3.44
4:     Calculate $B(G/P^{ex}(t))$                           ▷ Definition 3.3.51
5:     Calculate source $s$ and set $t \longleftarrow treeNode(s)$   ▷ Continue search in child/source
6: $j^* \longleftarrow t$                                    ▷ The leave is the source
```

The algorithm starts with the root of T in *Line 1*. In each iteration the extended quotient graph is calculated (*Line 3*). Then, the subbases for the extended quotient graph for the vertices of the not extended quotient ($V(G/P)$) is calculated in *Line 4*. Questioning the oracle about the basis elements reveals the source. Here, any vertex in the modules of the original graph corresponding to the basis vertices in the extended quotient can be questioned. The child of the current tree node corresponding to the source $treeNode(s)$ is the tree node where the search continues (if it is not a leave of the tree). If a leave is found, it corresponds to the single source node in the original graph.

Proposition 3.3.53 (Correctness of Algorithm 1)**.** *Algorithm 1 solves the* deterministic online source detection *problem.*

Proof. It is proven that, when the module represented by the current node t includes the true source, this is true also for the module represented by the child of t that corresponds to the source in the extended quotient of t. If the source is in the module $M(t)$ of the current tree node t, then there is a module $M' \subset M$ in the partition P given by the children of t, that includes the source. By Proposition 3.3.52 the source is uniquely attributable to the child representing the module including the source. Then, as this property holds for the root, it carries down to the leave of the tree, that is returned as source. □

Remark 3.3.54 (Implementation Algorithm 1). *Implementing Algorithm 1 one would reuse already questioned vertices in the current module and calculate corresponding restricted subbases. Also one would try to take oracle vertices from a module, that are as reusable as possible in future iterations, e.g., by always (recursively) choosing the vertex in a module that is included in the maximal submodule of this module, i.e., vertices corresponding to deepest branches in the modular decomposition tree.*

Also extended quotients shortest outer paths must not be recalculated in the original graph, but can, by Proposition 3.3.47, be calculated in the extended quotient of the parent (for the root the extended quotient equals the quotient graph).

For extended quotient calculations one would also shorten the shortest outer paths to only add one additional node and adjust edge lengths to keep the length of the shortest outer path.

The calculation of extended quotient bases might be impossible, then one would have to delete child nodes from the modular decomposition tree of the current node and make their children direct children of the current node.

3.4 Stochastic source detection

In the present work it is assumed that the distributions involved are known. If this is not the case the parameters of the distributions are estimated and then one can proceed as if this estimates would be true. In the context of source detection in continuous space already further work has been done [19].

In this subsection the general case with normally distributed random measurement errors ϵ_i is considered. The measurements r_k become random variables, and thus also the estimated parameters t_s, c, and $s \in V$. It may make sense to measure multiple times at a particular node.

Definition 3.4.1 (Stochastic Source Certificate). *For a given $\alpha \in (0,1)$ we call a node s the* probable source *of the spreading process, if t_s is finite and if a given statistical test passes with an error probability $1-(1-\alpha)^{(1/N)}$ for the hypothesis $t_s < t_j$ for all nodes j for which an edge (v_j, s) or $(s, v_j) \in E$. Here N is the number of edges $(v_j, s), (s, v_j) \in E$.*

3.4.1 Offline source detection

We start by looking at the source inversion (resolving) problem S3) from Definition 3.2.8.

Definition 3.4.2 (Source Estimator). *Given a multiset (nodes can be queried multiple times) of nodes $\widehat{S}$ and the corresponding oracle answers $r_{\widehat{S}}$, we define the source estimator similar to* (3.1) *as*

$$j^* := \arg\min_{j\in V} J^*_{j,\widehat{S}} := \arg\min_{j\in V} \min_{t_s,c\geq 0} J_{j,\widehat{S}}(t_s,c) \tag{3.4}$$

$$:= \arg\min_{j\in V} \min_{t_s,c\geq 0} \sum_{k\in\widehat{S}} (c\, d_{j,k} + t_s - r_k)^2 \tag{3.5}$$

as the most likely source for $\widehat{S}$ in a least squares sense.

For fixed source estimate $j \in V$, the solution of the linear regression problem (Definition 3.4.2) can be derived analytically [20, pages 4–5 for the unconstrained solution] as

$$c(j, \widehat{S}) = \max\left(0, \frac{\sum_{i \in \widehat{S}} (r_i - \bar{r})(d_{j,i} - \bar{d}(j))}{\sum_{i \in \widehat{S}} (d_{j,i} - \bar{d}(j))^2}\right), \tag{3.6}$$

$$t_s(j, \widehat{S}) = \bar{r} - c(j, \widehat{S})\ \bar{d}(j) \tag{3.7}$$

with $\bar{r} = \text{mean}(r_{\widehat{S}})$ and $\bar{d}(j) = \text{mean}(d_{j,\widehat{S}})$. This allows to evaluate $J^*_{j,\widehat{S}}$ for all $j \in V$ and derive an estimate j^* via enumeration, similar to the deterministic case. If the source can be resolved with a certain probability depends obviously on the choice of the multiset $\widehat{S}$.

To derive error estimates and oracle questioning node choosing strategies the estimator properties are investigated.

Definition 3.4.3 (Source estimator set). *Given a multiset (nodes can be queried multiple times) of nodes $\widehat{S}$ with $|\widehat{S}| = q$ the* source estimator set *for all nodes is*

$$\widetilde{A}_i = \{x \mid x \in R^q,\ \min_{t_s, c \geq 0} \sum_{k \in \widehat{S}} (c\, d_{i,k} + t_s - x_k)^2 \leq \min_{t_s, c \geq 0} \sum_{k \in \widehat{S}} (c\, d_{j,k} + t_s - x_k)^2\, \forall\ j \in V\},$$

i.e., all oracle answers x which would lead to i being the (or a) most likely source.

To help conceptualize the set and derive some properties of the source estimator set its center set is defined.

Definition 3.4.4 (Source estimator set center). *Given a multiset (nodes can be queried multiple times) of nodes $\widehat{S}$ with $|\widehat{S}| = q$ the* source estimator set center *for all nodes is*

$$C_i = \{x \mid x \in R^q, x = cd_{i\widehat{S}} + t_s, \forall\ c \geq 0, t_s \in R\}.$$

It includes all oracle answers x which would lead to i being the (or a) most likely source with perfect fit ($J_{i,\widehat{S}}(t_s, c) = 0$).

Basically half of a two dimensional subspace spanned by the vector $\mathbb{1}$ and the vector d_{iB} is including the set. As the velocity is positive one can cut the subspace along the subspace spanned by $\mathbb{1}$ and choose the half in which the vector d_{iB} points to get the center set. This center sets have the following properties.

Proposition 3.4.5 (Trivial source estimator set center intersection). *If $\widehat{S}$ is spread-resolving (Definition 3.3.7) the source estimator set centers intersect* only *in the one dimensional subspace spanned by $\mathbb{1}$.*

Proof. As $\widehat{S}$ is spread resolving no two vertices are spread equivalent, which is the same as to say there is no intersection between the set centers with positive c. The solution $c = 0$ is trivial. □

The intersection of the centers is the physically uninteresting case, where the oracle gave the same answer to all questions and infinite signal speed is estimated ($c = 0$), hence having no information about the source at all, i.e., all vertices are equally (un)likely.

Now the probability of source detection for the above estimator is derived.

Definition 3.4.6 (Source estimation probability)**.** *Given a spread-resolving multiset $\widehat{S}$, the true source $k \in V$ and the true parameters c, t_s as well as the σ the* source estimation probability *of a vertex i:*

$$P_{i,k}(c, t_0) = \int_{\widetilde{A}_i} z \exp(-\|d_{kB}/c + t_0 - x\|^2/(2\sigma^2))dx. \tag{3.8}$$

Where z is the normalization constant of the multidimensional Gaussian distribution density function.

Proposition 3.4.7. *For every source k the source estimation probabilities for all vertices sum to one, i.e., $\sum_{i\in V} P_{i,k}(c, t_0) = 1$.*

Proof. Let the source be any node k, with (unknown) speed and initial time. Then, the integration is performed over the density of a Gaussian distribution in the regions $\widetilde{A}_i$, $i \in V$. It suffices to show that the points contained in more than one of these sets have measure zero. The subspace spanned by $\mathbb{1}$ is contained in all of them, but has measure zero. As the source estimation set centers are distinct everywhere else, the intersections between the source estimator sets have measure zero everywhere. □

Based on this the oracle query placement problem S3) from Definition 3.2.8 can be solved. There are different approaches.

Definition 3.4.8 (Minimal estimation probability)**.** *Given the oracle variance σ the* minimal estimation probability *is*

$$P_{est} = \min_{i\in V} \int_{(0,\inf)\times\mathbb{R}} P_{i,i}(c, t_s)dcdt_s.$$

Definition 3.4.9 (Minimal number oracle question placement)**.** *Given the oracle variance σ and a minimal confidence level $\alpha \in (0, 1)$ the minimal cardinality multiset $\widehat{S}$ is given by*

$$\widehat{S} = \arg\min_{\widehat{S}} |\widehat{S}|, s.t. : P_{est} \geq 1 - \alpha.$$

Definition 3.4.10 (Maximal detection probability oracle question placement)**.** *Given the oracle variance σ and a maximal cardinality U the maximal minimal estimation probability multiset $\widehat{S}$ is given by*

$$\widehat{S} = \arg\max_{\widehat{S}} P_{est}, s.t. : |\widehat{S}| \leq U.$$

The same can be done for type II errors instead of type I errors (Definition 3.4.9).

Definition 3.4.11 (Maximal misestimation probability)**.** *Given the oracle variance σ the* maximal misestimation probability *is*

$$P_{est} = \max_{i\in V} \sum_{j\in V\setminus\{i\}} \int_{(0,\inf)\times\mathbb{R}} P_{i,j}(c, t_s)dcdt_s.$$

Also one could think of combining type I and II errors with the number of oracle queries in one criteria (e.g., minimizing queries while controlling error of type I+II). However all of this approaches suffer from the problem, that very difficult potentially high dimensional integrals have to be solved while enumerating over many different possible multisets. Hence practically this are challenging problems.

Additionally, even if such an offline search with controlled error strategy for oracle question placement is performed, the actual oracle answers can still be uninformative (e.g., close to boundaries of the source estimator sets) and therefore the most likely source does not meet the a priory chosen error bounds (type I and/or type II). This is a major drawback of the stochastic setup, and can not be changed.

Therefore the stochastic setup is inherently suited for an online approach, where in every iteration one can decide on termination based on the actual oracle answers and the information they include about the source.

3.4.2 Online source detection

To solve the source detection problem S3) from Definition 3.2.8 in the online setting we use the estimator from Definition 3.4.2. The estimator for the offline case, given a multiset of oracle queries to calculate a best fit source vertex once, can be used in every iteration during the online source detection.

As the true source estimation probabilities are difficult to calculate, we use a different approach.

Definition 3.4.12 (Stochastic Spread-Resolving Set). *Given $a, b \in \mathbb{R}_+$, a source estimate $j^* \in V$, a resolution radius $\gamma > 0$, and values $J^*_{j,\widehat{S}}$ $\forall j \in V$, we define*

$$B^{j^*}_{\gamma} = \{i \in V : \; d_{j*,i} \leq \gamma \} \tag{3.9}$$

and call the multiset $\widehat{S}$ stochastically spread-resolving *(SSR), if an F-Test is successful for a confidence α with*

$$\frac{\left(\min_{j \in V \setminus B^{j^*}_{\gamma}} J^*_{j,\widehat{S}}\right) - J^*_{j^*,\widehat{S}}}{J^*_{j^*,\widehat{S}}} \frac{b}{a} \geq F^{-1}_{a,b}(\alpha) \tag{3.10}$$

Note that this approach is heuristic, because the statistic is not F-distributed.

Example 8 (Stochastic Source Inversion). We consider our example graph with oracle queries at $\widehat{S} = \{1, 4, 6, 7, 9\}$ resulting in

$$r_{\widehat{S}} = (1.44327, 0.31493, 3.43784, 5.48041, 4.77700).$$

We can calculate best fit regression lines for all ten nodes:

node	c	t_s	objective value
0, 6, 7, 8, 9	$1e-10$	3.09069	19.09375
1	1.49215	0.40482	3.95344
2	2.12480	−1.15892	1.03461
3	3.39669	−5.06137	5.24874
4	1.65808	0.10614	0.39891
5	3.39669	−1.66468	5.24874

The smallest objective value is obtained for node 4. However, $\widehat{S}$ does not spread-resolve nodes 3 and 5 (e.g., for $t_s = c = 1$ we have $d_{3,k} = t_s + cd_{5,k}$), resulting in not distinguishable optimal solutions (objective value, c) with different t_s. For $a = b = 1$, $\alpha = 0.05$ and the ball $B^2_{1.5} = \{1, 2, 4\}$ the F-test fails with 12.158 and a cutoff value of 161.45. Thus $\widehat{S}$ is not SSR, and 4 is not a probable source, which is accurate as r_B was simulated for $s = 2, c = 2, t_s = -1$, and a standard deviation of 1.

For the experimental design problem S1) in Definition 3.2.8 we use A-optimality, i.e., we choose oracle queries that minimize the following function.

Definition 3.4.13 (Set Variance). *For a given multiset $\widehat{S}$, variances σ_j, and $\lambda \in [0,1]$ we define the* set variance

$$\Phi(\widehat{S}) := \sum_{j\in V} \lambda \frac{Var[c(j,\widehat{S})]}{\sigma_j^2} + (1-\lambda)\frac{Var[t_s(j,\widehat{S})]}{\sigma_j^2}, \tag{3.11}$$

calculated using the variances of the parameter estimates (3.6-3.7) according to [108, Section 2.4],

$$Var[c(j,\widehat{S})] = \frac{\sigma^2}{\sum_{i\in\widehat{S}}(d_{j,i}-\bar{d}_j)^2}$$

$$Var[t_s(j,\widehat{S})] = \frac{\sigma^2 \sum_{i\in\widehat{S}} d_{j,i}^2}{|\widehat{S}| \sum_{i\in\widehat{V}}(d_{j,i}-\bar{d}_j)^2}$$

with an unknown, but fixed σ^2.

To prove convergence, and also to avoid observed unwanted numerical behavior, we restrict the multiplicities of the multiset $\widehat{S}$. The number of queries per node must not differ by more than 1. This avoids that specific nodes are queried significantly more often than others.

Definition 3.4.14 (Feasible Oracle Queries). *Let V be given. A multiset $\widehat{S}$ of V is called* feasible, *if the multiplicities of all $i \in V$ within $\widehat{S}$ do not differ by more than 1. We denote by $V^{\widehat{S}}$ the subset of V containing all nodes that can be added to a feasible $\widehat{S}$ and maintain feasibility.*

Now, we can now formulate a source detection algorithm realizing Definition 3.2.8.

The goal of Algorithm 2 is to find a probable source j^* with a small number of oracle queries, assuming considerable practical costs (e.g., increased risk of side effects for intracardiac measurements). Concerning the computational complexity per iteration of Algorithm 2, the main calculations happen in Lines 5, 6, and 10. The inner optimization problems can be solved analytically, compare (3.6-3.7), with an effort proportional to $|V|$. This is similar to calculating the

Algorithm 2 Stochastic Source Detection

Input: Graph (V, E) with shortest distances d, access to oracle $\mathcal{V}$, parameters a, b, α, λ, variance weights σ_j

Output: Probable source j^*, SSR set $\widehat{S}$

```
1:  $i_1, i_2 \leftarrow \arg\min_{i_1 \neq i_2 \in V} \Phi(\{i_1, i_2\})$          ▷ See Def. 3.4.13
2:  $\widehat{S} \leftarrow \{i_1, i_2\}$                                          ▷ Initialize set $\widehat{S}$
3:  for $i$ in $3 \dots i_{\max}$ do
4:      $r_{\widehat{S}} \leftarrow \mathcal{V}(\widehat{S})$                      ▷ Update oracle $\mathcal{V}$ query
5:      Calculate $c(j, \widehat{S}), t_s(j, \widehat{S}) \ \forall\ j \in V$     ▷ See (3.6-3.7)
6:      Calculate objectives $J^*_{j,\widehat{S}}$ and $j^*$                       ▷ See (3.4-3.5)
7:      Calculate $\gamma = \frac{J^*_{j^*,\widehat{S}}}{(|\widehat{S}|-2)c(j^*,\widehat{S})}$   ▷ For SSR test
8:      if $\widehat{S}$ is SSR then                                               ▷ See (3.9-3.10)
9:          break
10:     $i^+ \leftarrow \arg\min_{j \in V^{\widehat{S}}} \Phi(\widehat{S} \cup \{j\})$   ▷ See Defs. (3.4.13-3.4.14)
11:     $\widehat{S} \leftarrow \widehat{S} \cup \{i^+\}$                           ▷ Add node to $\widehat{S}$
```

set variance in (3.11). The overall effort to evaluate all objective functions $J^*_{j,\widehat{S}}$ and minimizing over $V \setminus B^{j^*}_{\gamma}$ in Line 8 and over $V^{\widehat{S}}$ in Line 10 is then proportional to $|V|^2$, where clever look-up tables can be applied to increase performance. Note that the distance resolution in Line 7 is calculated by dividing the estimated standard deviation by the estimated slope $c(j^*, \widehat{S})$.

Given the general applicability of Algorithm 2 and the stochasticity of the task, we can not expect that the algorithm has a deterministic bound on the number of necessary iterations. However, the well-posedness follows from the following result.

Corollary 3.4.15 (Convergence in the limit)**.** *Assume we remove Lines 8–9 in Algorithm 2. Then there is an i_{max} such that the output of Algorithm 2 is $j^* = s$.*

Proof. In Line 6 of Algorithm 2 we calculate (3.4-3.5)

$$j^* = \arg\min_{j \in V} J^*_{j,\widehat{S}}.$$

We want to show that $j^* = s$, i.e., that

$$J^*_{j,\widehat{S}} = \min_{t_s, c \geq 0} \sum_{k \in \widehat{S}} (c\, d_{j,k} + t_s - r_k)^2$$

is smallest for $j = s$, if $\widehat{S}$ is large enough. As $\widehat{S}$ is augmented by one node in every iteration in Line 11, this correlates to a longer runtime and a larger $i_{\max}$.

We use Definition 3.2.4 and the true model for s for

$$r_k = d_{s,k} c + t_s + \epsilon_k$$

and the analytical solutions (3.6-3.7) to obtain

$$\begin{aligned}
J^*_{j,\widehat{S}} &= \sum_{k\in\widehat{S}} (c(j,\widehat{S})\, d_{j,k} + t_s(j,\widehat{S}) - r_k)^2 \\
&= \sum_{k\in\widehat{S}} (c(j,\widehat{S})\, d_{j,k} + \bar{r} - c(j,\widehat{S})\, \bar{d}(j) - r_k)^2 \\
&= \sum_{k\in\widehat{S}} (c(j,\widehat{S})\, (d_{j,k} - \bar{d}(j)) + (\bar{r} - r_k))^2 \\
&= \sum_{k\in\widehat{S}} \Big(c(j,\widehat{S})\, (d_{j,k} - \bar{d}(j)) \\
&\qquad -c(d_{s,k} - \bar{d}(s)) - (\epsilon_k - \bar{\epsilon})\Big)^2 .
\end{aligned}$$

We look at $c(j,\widehat{S})$ separately and use

$$f(x_{\widehat{S}}, j) = \frac{\sum_{i\in\widehat{S}}(x_i - \bar{x})(d_{j,i} - \bar{d}(j))}{\sum_{i\in\widehat{S}}(d_{j,i} - \bar{d}(j))^2}$$

as abbreviation:

$$\begin{aligned}
c(j,\widehat{S}) &= \max\left(0, f(r_{\widehat{S}}, j)\right) \\
&= \max\Big(0, c\, f(d_{s,\widehat{S}}, j) + f(\epsilon_{\widehat{S}}, j)\Big)
\end{aligned}$$

The term $\bar{\epsilon}$ in $f(\epsilon_{\widehat{S}}, j)$ is a Gaussian distribution $\mathcal{N}(0, \sigma^2/|\widehat{S}|)$. As the probability $P(|\bar{\epsilon}| < \gamma)$, $\gamma > 0$ tends towards one there is no influence of this term in the limit. Then $f(\epsilon_{\widehat{S}}, j)$ can be rewritten with notation $g_i = (d_{j,i} - \bar{d}(j))/\sum_{i\in\widehat{S}}(d_{j,i} - \bar{d}(j))^2$ as $\sum_{i\in\widehat{S}} \epsilon_i g_i \sim \mathcal{N}(0, \hat{\sigma}^2)$ with variance

$$\hat{\sigma}^2 = \sigma^2 \sum_{i\in\widehat{S}} g_i^2 = \sigma^2 / \sum_{i\in\widehat{S}} (d_{j,i} - \bar{d}(j))^2.$$

As also this Gaussian stochastically converges towards 0 (as $\bar{\epsilon}$ above), it has no influence. Inserting the remaining parts of $c(j,\widehat{S})$ back into the objective yields

$$J^*_{j,\widehat{S}} = \sum_{k\in\widehat{S}} \left(h_{j,k} - \epsilon_k\right)^2 = \sum_{k\in\widehat{S}} h_{j,k}^2 - 2h_{j,k}\epsilon_k + \epsilon_k^2.$$

As above, the term $\bar{\epsilon}$ is neglected because of its stochastic convergence to zero and we used

$$h_{j,k} = \max\Big(0, c\ f(d_{s,\widehat{S}}, j)\Big)\left(d_{j,k} - \bar{d}(j)\right) - c\left(d_{s,k} - \bar{d}(s)\right)$$

The difference between the true source objective and any other objective is in this term. Because $f(d_{s,\widehat{S}}, s) = 1$ we have $h_{s,k} = 0$.

For all other objectives the term $\sum_{k \in \widehat{S}} h_{j,k}^2$ grows at least linear in the size of $\widehat{S}$ because with V as spread-resolving set $\sum_{k \in V} h_{j,k}^2$ is bounded from below by a positive value and we add elements to $\widehat{S}$ in chunks of V.

The term $-2 \sum_{k \in \widehat{S}} h_{j,k} \epsilon_k$ is Gaussian $\mathcal{N}(0, \tilde{\sigma}^2)$ with variance $\tilde{\sigma}^2 = 4\sigma^2 \sum_{k \in \widehat{S}} h_{j,k}^2$ which grows at most linearly in the size of $\widehat{S}$ because $\sum_{k \in V} h_{j,k}^2$ is bounded from above.

The last term is $\widetilde{\chi}^2$ distributed with $|\widehat{S}|$ degrees of freedom. In the limit this tends to a Gaussian distribution with mean $|\widehat{S}|$ and variance $2|\widehat{S}|$. For the true source objective this is the only existing term.

Subtracting the true source objective from any other objective the result is Gaussian $\mathcal{N}(\mu, \hat{\sigma}^2)$ with $\mu = \sum_{k \in \widehat{S}} h_{j,k}^2$ and $\hat{\sigma}^2 = \tilde{\sigma}^2 + 2|\widehat{S}|$. The probability that it is greater than zero tends to one because the mean grows at least linearly and the variance grows at most linearly. □

4 | Medical application

We start with the medical background information for the non medical reader in Section 4.1.1. The following Sections of this chapter are from [107].

4.1 Medical background

The reader familiar with the medical background might skip this subsection.

4.1.1 Cardiovascular system

The description of the anatomy and the physiology here is based on [50]. A more detailed and thorough description can be found there. The cardiovascular system in humans is the transportation system of the body. It transports oxygen and nutrients as well as many other substances to all organs to sustain live. It was even seen so central to live that the indication of death was that the heart stopped beating [91]. The heart is the central part of the cardiovascular system supplying the energy to fulfill the transportation by pumping the blood through the arteries and veins. The blood is the liquid that actually carries the substances. Together the heart, the blood vessels (arteries and veins), and the blood form the circulatory system. Also the lymphatic system is part of the cardiovascular system but we will not consider it here.

We are interested in a specific disease of the heart, hence we will continue our description of the heart and do not take into account the blood itself or the blood vessels directly. Also we will focus on the key factors of the disease in contrast to normal heart function.

The heart consists of four chambers: right and left atrium, right and left ventricle. The right atrium and ventricle as well as the left atrium and ventricle form two pairs that work together to pump blood through part of the body to the other pair of chambers. The right pair is weaker because it only pumps the blood trough the lung to the left pair. The left pair is stronger and pumps the blood through the whole body. In both cases the blood first enters the atrium and is pumped from there to the bigger ventricle. The ventricle pumps the blood out of the heart. Back flow of blood is prevented by a valve at each chamber outlet. A chamber pumps blood by a concurrent contraction of its tissue. The coordination of the contraction in one chamber and between the chambers is achieved by the conduction system of the heart [7].

The conduction system of the heart triggers synchronized contraction of the heart muscle tissue by electrical activation. The activation is initiated by the sinus node. It is located on the top of the right atrium inside the atrium's wall. The sinus node autonomously produces and rhythmic electric activation. This activation then spreads over the walls (muscular tissue) of the atrium chambers and causes their contraction. Then the activation is blocked by the cardiac skeleton which isolates the atria from the ventricles. Only the atrioventricular node is conducting through this barrier. It is located right in the middle of the heart between all four chambers. It delays the electrical activation such that the blood from the atria can fill the ventricles before the ventricles contract. From the atrioventricular node the signal is conducted via the Bundle of His. The Bundle of His is a structure branching over the ventricles ensuring a fast conduction of the signal over the whole ventricle. Due to this fast pathway the activation can simultaneously reach from the bundle of His over the Purkinje fibers to the ventricular muscle tissue. Therefore, this tissue contracts together and pumps the blood out of the ventricles into the body.

Note that this is a very short and simplified description of the cardiovascular system, especially the cellular and detailed tissue level are not considered here. Also the upper organ level, regulation of heart activity, heart rate, pumping volume, regulation of blood flow through different parts of the body, especially the heart itself[1] are not described.

4.1.2 Premature ventricular beats

Premature ventricular beats (PVB) are a ventricular tachycardia in which a single source location causes arrhythmic off beats that break the sine rhythm. This source location acts like the sine node, causing too early contraction when the ventricles are not filled fully with blood yet. This off beats increase the heart rate but decrease the pumped blood volume. When they are frequent they can have serious impact on the patients health and need to be treated. In [1] PVBs are described as highly symptomatic, when patients have another heart disease and can even cause cardiomyopathy.

The treatment of PVBs is done via catheter ablation [90, 112]. The procedure is performed in two steps: First an catheter with an electric sensor at the tip is used to localize the source of the PVB. Then another catheter is used to ablate the source. This is usually done by heating or cooling of the tissue with the tip of the catheter. This forms a scar in the source region, which looses its electrical conduction properties and its ability to initiate PVBs.

The more time consuming part of the procedure is the source localization part. Here the doctor usually performs an heuristic search and measures at different locations the time difference for a PVB between the time when the signal passes the catheter tip and the time when the signal is recorded in the external ECG device. The largest found time difference, i.e. the earliest activation point is the unknown source. Especially when the PVBs do not occurs frequently under surgery, it is time consuming to take a single measurement. Therefore it is of high interest to reduce the number of measurements needed to perform the search. Here we use our graph based search algorithm.

[1] The muscular tissue of the heart is supplied with oxygen and nutrients like any other organ through arteries and veins from the outside because the muscular tissue is too thick to be supplied from the blood inside the chambers.

4.2 Introduction

Premature beats (PBs) are a common finding in patients with structural heart disease, but they can also occur in otherwise healthy individuals. In patients with drug refractory symptomatic PBs or frequent monomorphic ventricular PBs in patients with reduced left ventricular ejection fraction, catheter ablation is well indicated [90, 112]. Since the introduction by Gepstein et al [43], 3-dimensional (3D) electroanatomic mapping systems are increasingly applied to locate the exact site of origin of PBs. However, infrequent occurrence of PBs during the procedure can hinder the creation of a detailed activation map within an acceptable period of time, thereby limiting procedural success. In these particular cases, it may be reasonable to first reconstruct the anatomy of the heart chamber during sinus or paced rhythm, adding the local activation times (LATs) to the existing anatomical information within a second step as a so-called remap. Figure 4.1A exemplarily displays the electrocardiogram of a young patient with long QT syndrome suffering from recurrent episodes of torsade de pointes tachycardia triggered by short-coupled left ventricular PBs. In this particular case, the operator decided to first reconstruct left ventricular anatomy. By obtaining LAT measurements (exemplarily displayed in Figure 4.1B for 3 [image I], 5 [image II], 7 [image III], and 27 [image IV] LAT measurements) at different locations within the previously generated geometry (remap), the site of origin could be identified (Figure 4.1B$_{\text{II-IV}}$).

Motivated by established systematic search routines as, for example, applied for the rescue of avalanche victims, we strived for developing an algorithm for optimized data acquisition to accelerate the mapping procedure in cases of rare arrhythmia occurrence [100]. Our strategy was based on the assumption that when deciding about the localization of each next LAT measurement, the amount of additional information may largely differ depending on the exact location of the measuring point. We therefore developed a mapping algorithm that is able to calculate the amount of additive value at each nodal point of the geometry and automatically position the next LAT measurement at the site of maximum additive information. Furthermore, the algorithm is able to predict earliest activation by extrapolation on the basis of the acquired LAT measurements with high accuracy.

4.3 Methods

4.3.1 Electrophysiological procedures and data acquisition

Seventeen patients who underwent catheter ablation of focal arrhythmias guided by a 3D mapping system (CARTO 3, Biosense Webster Inc., Diamond Bar, CA) between March 1, 2014 and August 31, 2015 were selected retrospectively from our database. The study was approved by the local ethics committee and was performed in accordance with the Declaration of Helsinki (64th WMA General Assembly, Fortaleza, Brazil, 2013). Data were recorded and analyzed using the LABSYSTEM PRO electrophysiological recording system (Boston Scientific, Marborough, MA,). Electroanatomic maps were established with the point-by-point acquisition mode of the CARTO system.

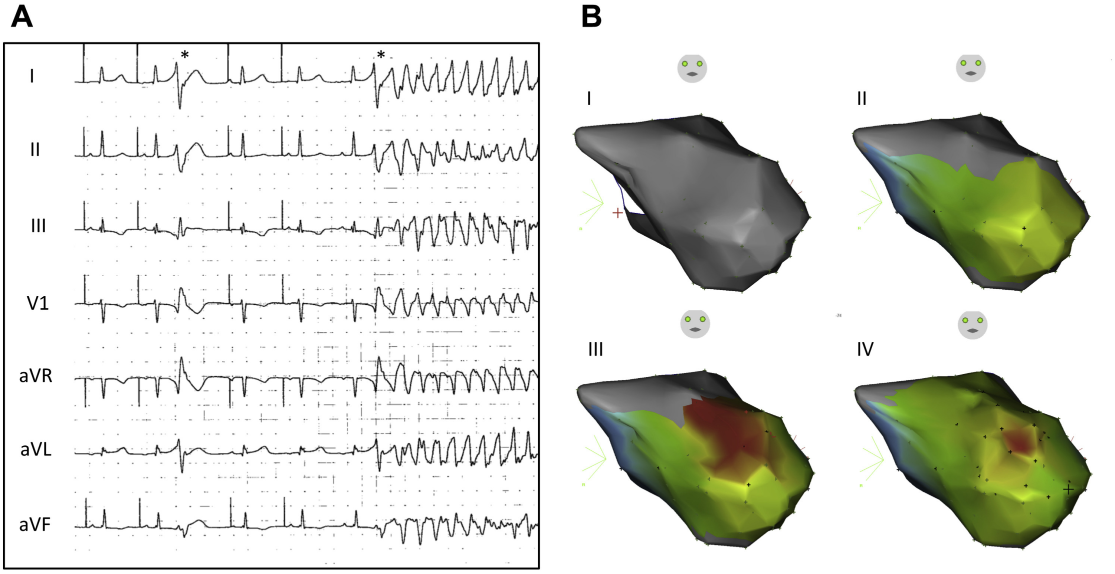


Figure 4.1: Electroanatomic mapping of short-coupled ventricular premature beats triggering TdP tachycardia. **A:** ECG of a young female patient with long QT syndrome suffering from recurrent episodes of TdP tachycardia triggered by monomorphic ventricular PBs (denoted by asterisk). Atrial pacemaker stimulation is highlighted by built-in pacemaker detection. **B:** Because of the infrequent occurrence of PBs during the ablation procedure, an anatomical geometry was first established (image B_{I}). Excitation propagation during PBs was then analyzed within the previously established anatomic map by point-by-point acquisition of LATs (images $B_{II\text{-}IV}$). The displayed remaps are based on the spatiotemporal information of 3 (image I), 5 (image II), 7 (image III), and 27 (image IV) mapping points. LATs are color coded, with red representing early activation times and blue late activation times. The site of successful ablation (image B_{IV}, red region) was located within the Purkinje system of the anteroseptal midventricular segment of the left ventricle. ECG = electrocardiogram; LAT = local activation time; PB = premature beat; TdP = torsade de pointes.

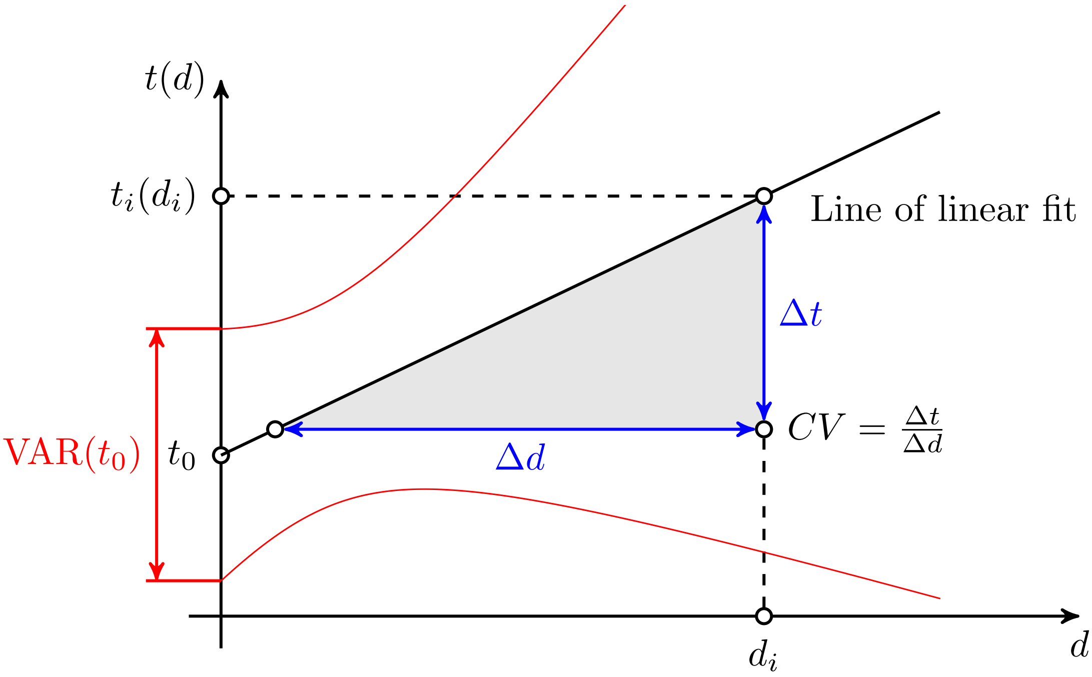


Figure 4.2: *Supplementary Figure: Schematic plot of regression analysis.*
This simple Figure visualizes the terms used in Formula 4.1. When performing regression analysis for a nodal point, the distance (d) between this point and any other mapping point (i) is plotted on the x-axis. The corresponding LAT is plotted on the y-axis (t_i). Based on distances and LATs of all mapping points, a linear fit could be calculated. The y-intercept (t_0) represents the LAT at the source of excitation. Every point, including the point at $d = 0$ (y-intercept), exhibits a certain variance (VAR) (red lines). Conduction velocity (CV) can be obtained from the linear fit by dividing time by distance. CV=conduction velocity, d=distance, LAT=local activation time, t=time, VAR=variance.

4.3.2 Development of patient-specific geometric models

For each patient, the raw data of the geometry of the heart chamber exported from the CARTO system consist of a triangular mesh. Some points (measurement points) are further labeled with a LAT obtained by the operator. On the basis of the spatiotemporal information exported from the CARTO system, we established a 3D geometry for each patient using the MATLAB software package (MathWorks, Natick, MA). Figure 4.3A schematically displays the structure of the surface of this mesh/graph. Within this graph, the distance between 2 nodal points was measured using the shortest distance over the connecting edges. The shortest distance between the 2 nodal points i and j is denoted as d_{ij}.

Considering the focal character of the arrhythmia, we assumed that electric excitation would spread centrifugally over the connecting edges of the mesh geometry with a constant conduction velocity (CV). CV for each patient was calculated on the basis of the acquired LATs and the geometric positions of the mapping points (Table 4.1). When fitting a straight line through these data, CV could be simply obtained from the slope of the line (Supplemental Figure 4.2). The LAT at a certain nodal point (t_i) is given by a linear model using the shortest path distances and the following Formula:

$$t_i = \frac{d_{ik}}{\mathrm{CV}} + t_0 - \mathrm{SEM} \tag{4.1}$$

where t_i is the arrival time of the signal at point i, d_{ik} is the shortest distance between nodal point i and source k, CV is the conduction velocity, t_0 is the unknown earliest activation time, and SEM is an individual measurement error. This error had to be included because the estimated CV obtained from the exported nodal points exhibited a certain degree of uncertainty, reflected by the r^2 value (Table 4.1).

Table 4.1: Correlation coefficients, estimated CVs, and analyzed heart chambers of all patients (CV = conduction velocity; LA = left atrium; LV = left ventricle; RA = right atrium; RV = right ventricle).

Patient no.	r^2	CV (m/s)	Heart chamber
1	0.86	1.8	LV
2	0.74	1.1	RA
3	0.81	1.4	LV
4	0.73	1.1	LA
5	0.56	1.8	LV
6	0.85	1.0	RV
7	0.81	1.3	LV
8	0.96	0.73	LV
9	0.61	1.2	RA
10	0.63	2.0	RV
11	0.90	0.85	LV
12	0.57	0.83	RV
13	0.80	1.1	LA
14	0.73	1.0	RV
15	0.77	0.93	RV
16	0.54	1.4	RV
17	0.64	1.6	LV

4.3.3 Prediction of earliest activation

When the algorithm obtained a measurement within the previously generated geometric model, the LAT at this specific nodal point was computed by the distance to the origin and the CV using Formula 4.1. On the basis of the location and LAT of the obtained measurements, the algorithm predicted the origin of the signal by solving a linear regression problem for every nodal point of geometry j. To estimate the CV and the initial time t_0, we minimized the objective

$$J_i = \sum_{i=1}^{n} \frac{d_{ik}}{\mathrm{CV}} + t_0 - t_i$$

Solving this problem for every nodal point on the geometry, the nodal point exhibiting the best fit (highest r^2 value) was considered the origin of the signal or at least the best estimate of the origin on the basis of the available information. An example is considering that a number of mapping points have been acquired and the algorithm tries to identify the site of earliest activation. In this case, it would go through all other nodal points each time establishing a regression analysis (see Supplemental Figure 4.2) with all available mapping points. Figure 4.3 explains the search routine within a simple planar mesh geometry. The red nodal point marks the true origin of excitation, and the black points (labeled a, b, and c) mark the measurement points already obtained at this time point. When assuming the blue point as the possible source, the calculated distances to the measurements points are 1, 3, and 3 numbers of edges. However, as the LATs at the measurement points in fact characterize the distance to the true origin (red point), there is no good correlation between distance and time (Figure 4.3C). In contrast, when assuming the red point as the possible source, the distances as well as the time delays to the measurement points (black points) are 1, 2, and 3. Figure 4.3D displays the excellent correlation of these points, thereby identifying the red point as true origin.

4.3.4 Optimizing the localization of the next measurement point

Using the aforementioned strategy, the algorithm is able to predict the most likely localization of the origin. However, a main goal of our work was to develop an algorithm that considerably reduces the number of measurement points. For this purpose, the algorithm needs to select that next measurement point that increases the quality of our estimate best. To identify the optimal next measurement point, the algorithm tried to minimize the standard error of the y-intercept of the regression curves for all nodal point using the following formula:

$$\sum_{l=1}^{N} \frac{a_l \mathrm{VAR}(t_{0l})}{\sigma^2} = \frac{a_l \sum_{i=1}^{n} d_{il}^2}{\sum_{i=1}^{n} (d_{il} - \bar{d}_l)^2}$$

where a_l is the positive weight used to define how important a nodal point and its detection as source is. The second regression parameter of the regression at nodal point i is t_{0i}. In other words, we wanted to minimize the variance of the y-intercept for all regressions we perform by picking the next measurement point. The reason for this approach is that every point of the regression curve possesses a certain degree of uncertainty. This is reflected by the variance (see red lines in Supplemental Figure 4.2). In addition to the correlation coefficient, the variance of the y-intercept (at distance 0) provides information about the quality of fit. Therefore, to identify the specific nodal point where a next mapping point would best improve the quality of the map, the algorithm again went through all nodal points checking how much a measurement at this specific point would reduce the variance of the y-intercept of the regression curves at all other points. This specific nodal point that results in the maximum reduction of the variance of the y-intercept of all other nodal points was identified as the next mapping point. The mean

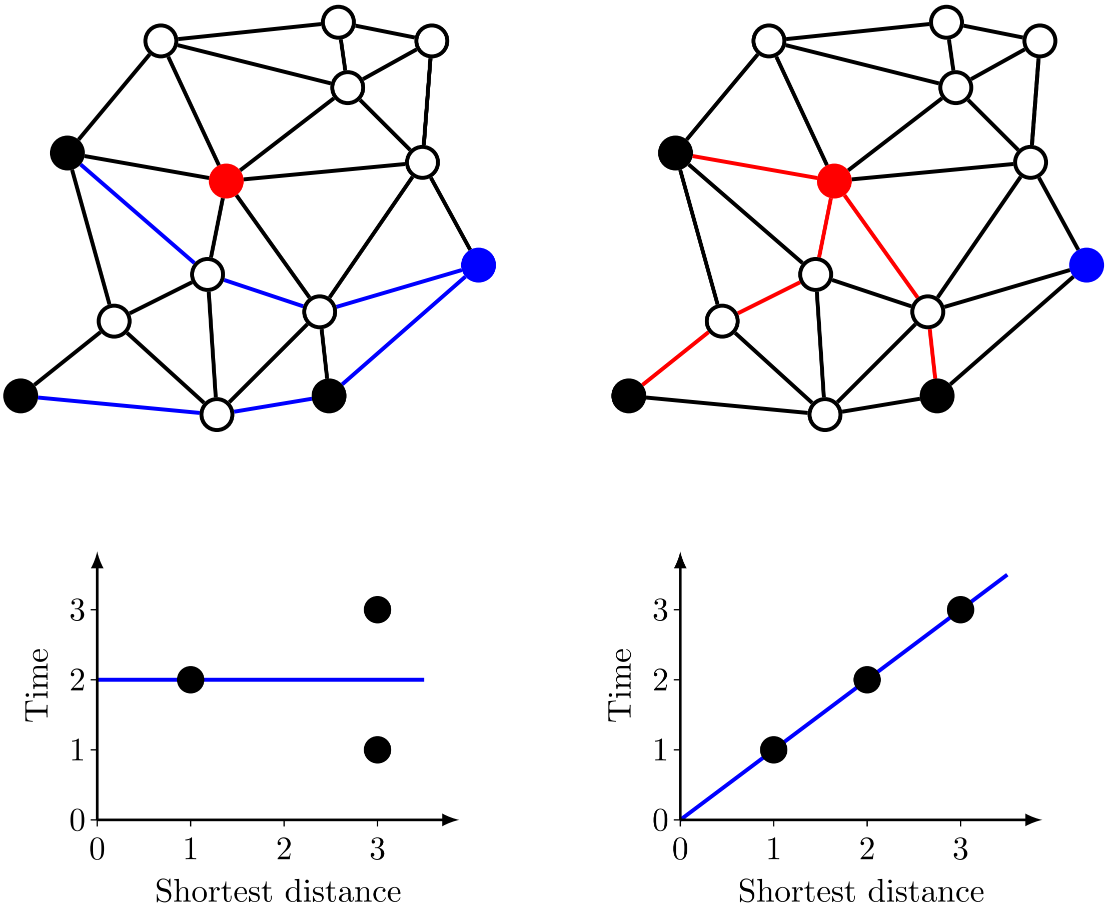


Figure 4.3: Schematic illustration of regression analysis within a simplified mesh graph. On the basis of the electroanatomic maps exported from the CARTO system, simplified mesh geometries were established. **A:** Within this simple example, the red point represents the true site of origin of a focally spreading electrical activation. The black points (labeled a, b, and c) represent nodal points at which LAT measurements have been performed. Considering a time delay of 1 arbitrary unit for the conduction from one to the next nodal point, the LATs at the measurement points a, b, and c are 1, 2, and 3, respectively (equivalent to the distance to the red point). **C:** Not knowing the site of origin and questioning whether the blue point might possibly be the source of the arrhythmia, one could draw a simple graph plotting the distance between the blue and black points (3, 1, and 3) in correspondence to the LATs (1, 2, and 3). Regression analysis within this plot reveals no correlation, thereby excluding the blue point as the true site of origin. **B and D:** When performing the same analysis for the red point, the graph reveals an excellent correlation, thereby identifying the red point as the true site of origin. LAT = local activation time.

overall computing time for the calculation of the optimal position of the next mapping point as well as the revised prediction of earliest activation was 124 ms.

4.3.5 Detecting the source

In order to locate the site of earliest activation, the algorithm, at a certain point in time, stops taking mapping points based on the degree of maximum additive information and starts localizing the site of origin by a direct search close to the predicted site of origin. This change of the search strategy seems reasonable, as points collected on the basis of maximum additive information are generally remote and not close to the predicted origin. To identify the point in time, when this second phase of the search routine needed to be entered, the algorithm continuously compared the quality of fit (derived from regression analysis) between the predicted site of origin and all points within 1 cm around it. As soon as the difference of the quality of fit to the surrounding points reached a certain threshold (surrounding points significantly worse), the search routine was changed to a more or less heuristic search around the predicted site of origin.

4.4 Results

4.4.1 Mapping of earliest activation by the operator

Electroanatomic maps from 17 patients who underwent ablation of focal arrhythmias were selected retrospectively from our database. On average, a number of 55.1 6 8.8 (n 5 17) LAT measurements had been acquired by the operator before ablation. The geometry of the aforementioned patient with long QT syndrome is displayed exemplarily in Figure 4.4A. The LAT measurements obtained by the operator are displayed as color-coded points, with red representing early activation times and purple late activation times (see timescale). The projection of the search path followed by the operator is displayed. Figure 4.4B displays the distance between the measuring points and the origin in relationship to the corresponding LATs. In this particular case, a total number of 27 LAT measurements were obtained. The corresponding linear regression exhibited a good correlation, yielding an r 2 value of 0.86 (slope 0.55 ± 0.044 ms/ mm; y-intercept 271.1 ± 1.1 ms; n = 27). Table 4.1 gives an overview of the correlation coefficients, the calculated CVs, and the mapped heart chambers of all patients.

4.4.2 Mapping of earliest activation by the algorithm

Next, we analyzed mapping performance of the developed algorithm. For this purpose, realistic models of excitation propagation had to be first established for each patient. We therefore computed the LAT for each nodal point of the mesh geometry using the electroanatomic map created by the operator, the identified origin, the calculated CV, and the individual measurement error. These geometries served as a test environment for the algorithm. At the beginning of the automated mapping procedure, the algorithm had information only about the anatomical situation derived from the mesh geometry, such as the exact localization of the nodal points and the connecting edges, which means that the algorithm was blinded to all LATs of the geometry.

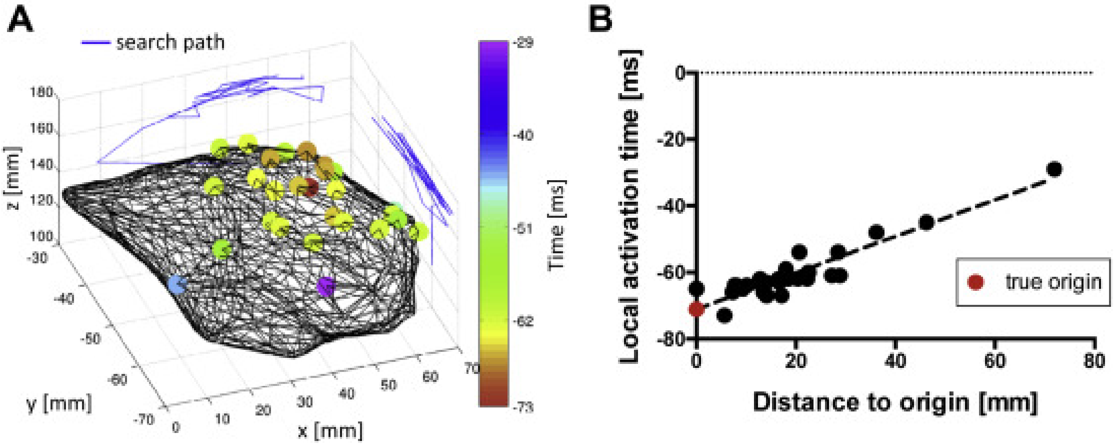


Figure 4.4: Evaluation of the mapping approach of the operator. **A:** Reconstructed mesh geometry obtained from the ablation procedure in the aforementioned patient with long QT syndrome. LAT measurement points obtained by the operator are displayed as points with color-coded activation times, with red representing early activation times and purple late activation times (see color scale). The search path from the LAT measurement to the LAT measurement followed by the operator is displayed as a projection on the frontal and sagittal planes. **B:** Distance between measurement points and the corresponding LATs shows a good correlation when assuming the site of best fit as the true origin (red point). LAT = local activation time.

At the very beginning of the automated mapping procedure, the first 3 LAT measurements were chosen en bloc because the selection of only 2 points would result in a perfect linear regression fit, rendering all points equally likely to be the origin. These 3 measurements are automatically performed by the algorithm purely on the basis of the shape of the geometry. Now that the LAT and localization of the initial 3 measurement points had been obtained, the algorithm went through all nodal points of the geometry, each time calculating the distance to the 3 initial measurement points and the linear regression for distance and LAT. The nodal point with the best fit was then assumed to be the origin (predicted origin). Figure 4.5A exemplarily displays the situation for the previously described patient (Figure 4.1) at this exact point in time. The initial 3 measurement points are displayed as black points. The quality of correlation (r^2) of each nodal point is color coded, with blue representing strong correlation and white weak correlation. The nodal point with the best correlation (predicted origin) is highlighted as a large blue point. The red point indicates the localization of the true origin. Since only 3 LAT measurements had been performed at this point in time, the predicted origin is still remarkably displaced from the true origin. The correlation between the distance to the measurement points and the LATs for the predicted origin (large blue point) is displayed in Figure 4.5B.

Next, the algorithm aimed at identifying that specific nodal point that would add maximum information about the localization of the true origin. Considering that the y-intercept of the regression curve represents the predicted origin, the algorithm searched for that specific nodal point at which an LAT measurement would reduce the variance of the y-intercept for the re-

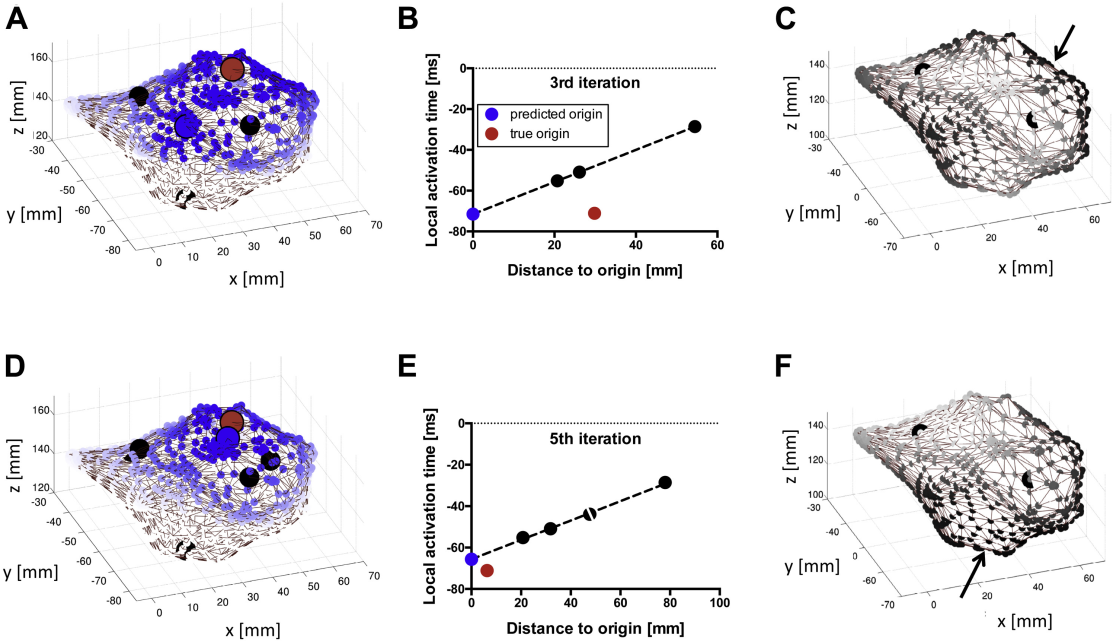


Figure 4.5: Mapping approach of the automated algorithm. **A:** The situation of the automated mapping algorithm at the time point of the third iteration in our exemplary simulation. Obtained LAT measurements are displayed as black points, with 1 measurement point hidden on the backside of the geometry. Nodal points of the geometry are color coded, with white representing unlikely sites of the predicted origin and blue likely sites of the predicted origin. The blue point represents the best site of the predicted origin based on the 3 LAT measurements obtained so far, and the red point represents the true origin. **B:** Regression analysis for the predicted origin (nodal point with the best available regression coefficient). **C:** Based on the measurements obtained so far, the additive value at each nodal point was calculated and visualized, with white representing low additive value and black high additive value. The location of maximum additive information (indicated by arrow) is selected as the next measurement point. **D–F:** The situation at the time point of the fifth iteration. Of note, the split point represents 2 measurements at different locations with the same distance to the origin and the same LAT. LAT = local activation time.

gression curves of all nodal points most. Figure 4.5C displays a mesh geometry showing the additive value at each nodal point, with black representing maximum additive information and white minimum additive information. The location of the maximum additive value is marked by an arrow (Figure 4.5C). Two iterations later and now based on the information of 5 LAT measurements, the predicted origin evidently migrates toward the true origin (Figures 4.5D–F). Figure 4.6A displays the situation after the seventh iteration when the algorithm located the true origin by placing the last measurement in this position. Comparable to Figure 4.4A, the acquired measurement points are color coded, with red representing early activation times and purple late activation times. Again, the search path is displayed on the frontal and sagittal planes. In contrast to the operator (27 measurements), the algorithm identified the true origin with 7 LAT measurements. Figure 4.6B displays the correlation between the distance to the origin and the LAT. To allow a direct comparison between the map automatically generated by our algorithm with the original CARTO map, we created a color-coded CARTO-like activation map (Figure 4.6C). Identically to the remap displayed in Figure 4.1B$_{III}$, our map is based on the spatiotemporal information of 7 LAT measurements. A direct comparison of both maps visualizes the obvious difference in accuracy.

4.4.3 Comparison between the operator and the algorithm

Next, we analyzed the diagnostic performance of the algorithm within all previously generated test geometries. Figure 4.7A displays the mean number of iterations that were necessary to identify the origin of the arrhythmia in each patient. Overall, the mean number of LAT measurements that were needed by the algorithm to identify the origin was 10 ± 0.51 ($n = 17$). Compared to the algorithm, the operator, on average, took 5 times as many LAT measurements (55 ± 8.8; $n = 17$; $P < .0001$). However, when performing a head-to-head comparison between the mapping performance of the operator and the algorithm, it has to be taken into account that the algorithm always started mapping within an existing anatomy whereas the operator in some cases had reconstructed the anatomy simultaneously while mapping activation times. For this purpose, Figure 4.7B compares only those particular cases in which the operator was able to map activation times within a preexisting anatomy (remap). In these 10 cases, the site of origin could be identified within 11 ± 0.89 LAT measurement points by the algorithm as compared to 42 ± 7.0 LAT measurement points by the operator ($n = 10$; $P < .001$).

4.5 Discussion

4.5.1 Clinical implications

The algorithm outperformed the operator in almost every case in terms of the number of mapping points necessary to locate the site of origin, thereby pointing to shorter procedure times. In our opinion, there are 2 main reasons for this observed effect: (1) When directly comparing the number needed based on iterative linear regression analyses, our algorithm is able to calculate the degree of redundancy from each nodal point of the geometry, thereby identifying the nodal point with maximum additive information. This optimized search routine guarantees that the site of earliest activation can be located with a low number of LAT measurements. (2)

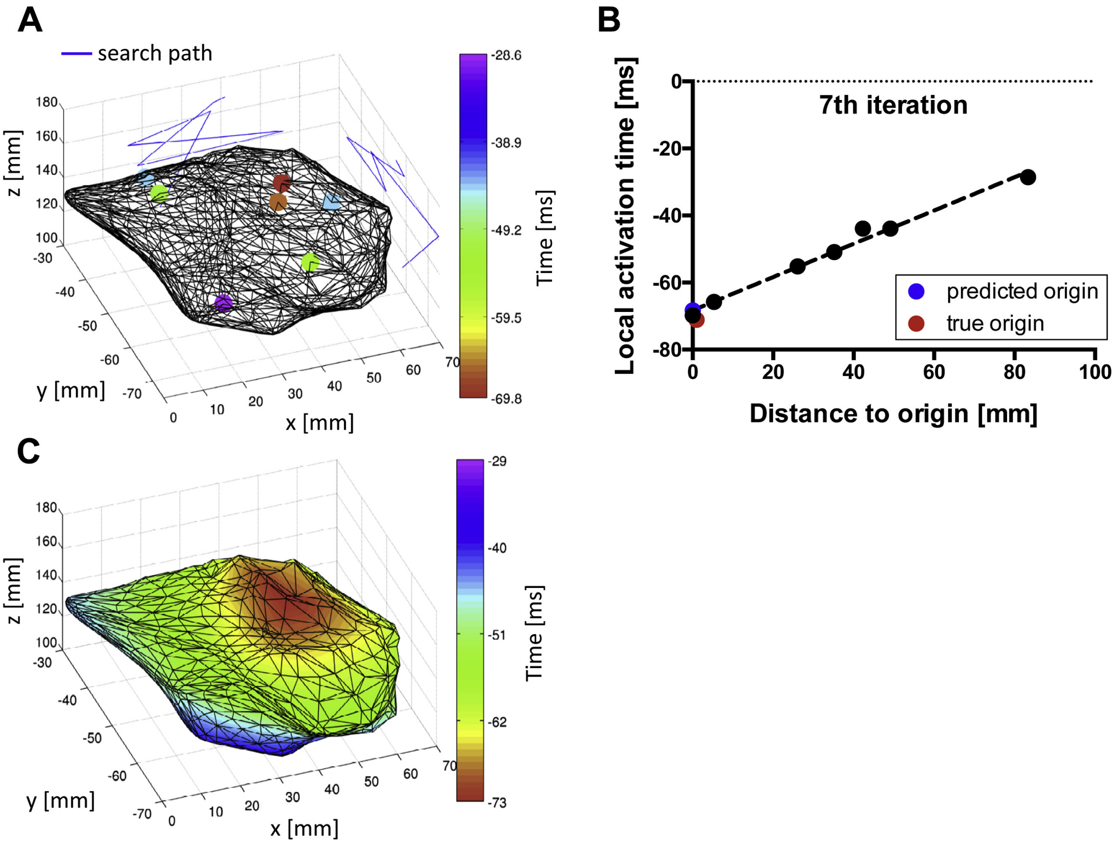


Figure 4.6: Evaluation of the mapping approach of the algorithm. **A:** Within the simplified geometry of the aforementioned patient, the algorithm identified the exact site of the true origin within 7 iterations. Again, the LAT measurements are displayed as color-coded points and the search path is displayed as a projection. **B:** Regression analysis of the 7 LAT measurements. **C:** To allow a direct comparison with the CARTO map based on the spatiotemporal information of the same number of LAT measurements (see Figure 4.1B$_{\text{III}}$), a CARTO-like activation map was established, with red representing early activation times and purple late activation times. LAT = local activation time.

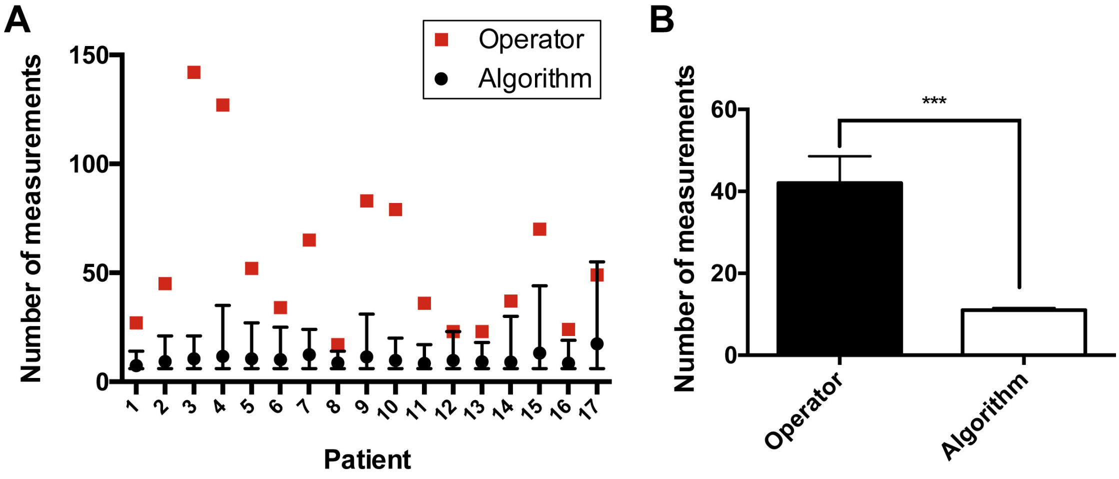


Figure 4.7: Systematic comparison of the mapping performance between the algorithm and the operator. **A:** Mean values and standard error of the LAT measurements obtained by the automated algorithm within 100 test runs per patient (displayed as black points). The number of LAT measurements taken by the operator are displayed as red squares. **B:** Compared to a mean number of 42 ± 7.0 LAT measurements taken by the operator in 10 patients with a remap, the algorithm was able to locate the site of earliest activation within 11 ± 0.89 LAT measurement points. LAT = local activation time.

Except for optimized data acquisition, our algorithm exhibits a second feature that might contribute to the significant reduction of mapping points. The linear regression analyses calculated for each nodal point are used not only to calculate the amount of additive information but also to predict the site of earliest activation. This means that LATs of any nodal point is extrapolated from the acquired LATs. In contrast, activation maps generated by the CARTO system are based on interpolation between the acquired mapping points. This difference might account for the higher quality of the activation map generated by our algorithm. As an example, when comparing the CARTO map based on the spatiotemporal information of 7 LAT measurements displayed in Figure 4.1B$_{\mathrm{III}}$ with our map based on the same number of LAT measurements (Figure 4.5C), a clear difference in the accuracy is visible. Interestingly, a quite similar approach using regression analyses has been published recently for the identification of the source of network-driven contagion phenomena such as the 2009 H1N1 influenza pandemic or the 2003 severe acute respiratory syndrome epidemic [24]. In their work the authors report that on the basis of geographical locations, arrival times of infection, and predicted traveling times, the source of the infection can be located using correlation analyses. Compared to our simple 3D electroanatomic model, these simulations are quite complex because of the inhomogeneous network structure and limited knowledge about the exact traveling times [52]. However, motivated by the good results of these studies, it might be a promising approach to further increase the degree of complexity of our models, for example, by including anatomical structures (ie, mitral or tricuspid annulus) or by including areas of scared tissue with reduced CV, for example, to allow fast activation mapping in scar-related ventricular tachycardia.

4.5.2 Alternative mapping techniques

Mapping of focally spreading cardiac arrhythmias is a relevant clinical topic, and a variety of strategies have been developed to accelerate localization, thereby increasing success rates. Hocini and coworkers [90, 112, 48] recently published promising data from a multicenter study applying a novel technique of high-resolution noninvasive mapping for the ablation of focal arrhythmia. However, it has to be taken into account that this technique requires a thoracic computed tomography scan, an electrode vest, and an additional mapping system (ECVue, CardioInsight Technologies, Inc., Cleveland, OH). A similar approach of inverse potential mapping based on the combination of a magnetic resonance imaging scan and a body surface potential map has recently been published by Bhagirath and coworkers [43, 18]. However, noninvasive mapping techniques have not yet entered clinical routine. Similar to noninvasive mapping techniques, the use of multielectrode catheters has also been proposed to accelerate endocardial mapping of focal arrhythmias. Using noncontact [100, 113] or contact [90, 95, 2] acquisition of endocardial electrograms, these systems allow simultaneous assessment of a larger number of LATs using basket or balloon catheters. Compared to all technologies described above, our algorithm offers the advantage that it may easily be implemented into standard 3D electroanatomic mapping systems without any need for additional hardware components or preprocedural imaging modalities.

4.5.3 Studi limitations

Because of obvious technical reasons, the diagnostic performance of the algorithm could be assessed only retrospectively using electroanatomic maps that had been exported from the CARTO system. Furthermore, operators with different degrees of expertise conducted the mapping procedures. Therefore, it can be suspected that fewer LAT measurements might have been necessary when the operator with most experience would have performed all procedures. However, in this way, the results of the study might even be more representative for a real-world situation.

4.5.4 Conclusion

We developed an automated mapping algorithm for the identification of the site of earliest activation within 3D electroanatomic maps. We further show that when compared to an operator, the algorithm is able to locate the site of earliest activation with a significantly lower number of LAT measurements. When integrated into an electroanatomic mapping system, this algorithm might significantly accelerate the procedure by guiding the operator to the optimal position for the next LAT measurement, thereby reducing the number of points with a high degree of redundant information. Furthermore, the algorithm would be able to predict the site of origin with high accuracy early during the mapping procedure.

5 | Numerical results

5.1 Implementation

We have implemented Algorithm 2 in `octave 5.2.0` [34]. The code is available with a permissive license in the GitHub repository `https://github.com/TobiasWeber/IMLR/`.

The implementation is a set of `octave` functions and scripts that should work in any `octave` installation with the `statistics` package. For Algorithm 2 only core `octave` was used. Data structures for graph representation were taken from the `octave` network toolbox [22], where also simple graph information and manipulation algorithms can be found. However, the algorithm works standalone as we implemented a different shortest path algorithm for efficiency reasons. It is closely related to the fast matrix multiplication shortest path algorithms and more suited to the `octave` programming language than the Dijkstra algorithm in the toolbox. For the numerical random scenarios and errors we use the statistics package of `octave`.

Remark 5.1.1 (Treating infinities)**.** *There are two different sources of infinite values.*

For directed graphs that are not strongly connected or graphs that are not connected, some pairs of nodes might have no shortest path between them, or just in one direction. The infinity pattern can be exploited to find the source by a clustering into connected subgraphs. These subgraphs can be used in step S3), and determined in an extra run of S1) by replacing infinity by 1 *and finite values by* 0*.*

Also the variances $Var[c(j,\widehat{S})]$ *and* $Var[t_s(j,\widehat{S})]$ *in Def. 3.4.13 may be infinite. As we are minimizing, this is not a problem though, if implemented carefully. If all variances in* $V^{\widehat{S}}$ *are infinite, we "minimize" by counting the non finite values in the sum over the variances and choose the "solution" with the least infinities (or NaNs).*

In the following and if not stated otherwise, we use hyperparameters $\alpha = 0.05$, $a = 1$, and $b = |\widehat{S}| - 4$ (see Def. 3.4.12) and $\lambda = 0$ and $\sigma_j = 1 + J^*_{j,\widehat{S}}$ (see Def. 3.4.13). Here, the σ_j were chosen to have larger weight on nodes with a smaller objective function value.

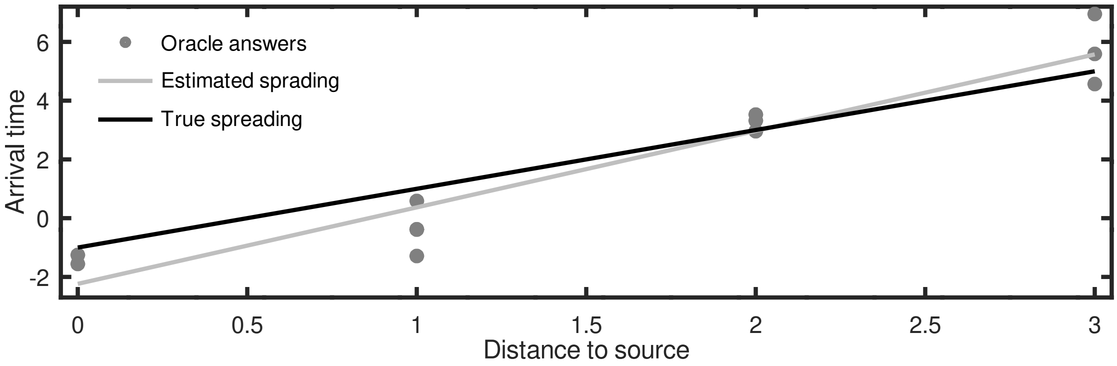


Figure 5.1: Regression after iteration 12 at the true (and estimated) source node s. Despite significant outliers, the estimated spreading approximates the true spreading quite well.

5.2 Illustration on Example Graph

We use our example graph with the same spreading process as in Example 8 ($s = 2, c = 2, t_s = -1$) to illustrate the behavior of Algorithm 2. The output for an instance with "average" behavior in terms of iteration count is as follows.

iter	i^+	j^*	$c(j^*, \widehat{S})$	$t_s(j^*, \widehat{S})$	$\frac{J^*_{j^*,\widehat{S}}}{\lvert\widehat{S}\rvert}$	$\min\limits_{j \in V \setminus B^{j^*}_Y} \frac{J^*_{j,\widehat{S}}}{\lvert\widehat{S}\rvert}$	α^*
1, 2	0, 5	5	3.34	-0.38	0.00	0.00	1.0000
3	1	2	3.79	-4.62	0.14	0.14	1.0000
4	7	1	2.09	-1.67	0.25	0.32	1.0000
5	2	4	2.97	-3.89	0.37	0.55	0.6137
6	4	2	2.25	-2.19	0.52	0.72	0.4716
7	8	2	2.44	-2.34	0.53	0.72	0.3800
8	3	2	2.48	-2.30	0.53	0.70	0.3255
9	9	2	2.71	-2.52	0.74	0.90	0.3434
10	6	2	2.72	-2.48	0.70	0.85	0.3054
11	2	2	2.56	-2.10	0.73	0.95	0.1905
12	1	2	2.60	-2.24	0.72	1.21	0.0494

The initialization in Line 1 results in $\{i_1, i_2\} = \{0, 5\}$. Until iteration 10 all nodes are selected once. In iterations 11 and 12 a second query at nodes 2 and 1 results in objective function values that are far enough apart such that the heuristic termination criterion (3.10) is fulfilled. In this instance the feasibility requirement in Def. 3.4.14 leads to oracle queries that might not be necessary. The last column depicts the converging α^* value, obtained by evaluating (3.10) (stopping criterion is that α^* is below 0.05). The parameters $c(j^*, \widehat{S})$ and $t_s(j^*, \widehat{S})$ converge slowly towards the real values and are still inaccurate at termination. The resulting regression line is a good fit for the measurements, though, compare Figure 5.1.

In summary, the strict termination criterion and feasibility requirement for oracle queries seem to be robust and avoid early termination, even when by chance a good fit is achieved as in the iterations 2 and 3.

5.3 Problem Instances

We used graph instances that we collected in three sets.
Col. The first set is an operations research library [11] with 30 instances from [40] and 79 DIMACS graph coloring instances [54] from different sources, e.g., [53, 70].
Misc. The second set comprises miscellaneous instances. It contains three simple water network instances from the `epanet` software for modeling of water networks [92], eight train networks from the `lintim` software [3], five instances from Mark Newman's webpage [85] (original sources [105, 75, 111, 62]), seven instances from the Weizmann laboratory collection of complex networks [5, 77, 82, 81], and 41 instances from the Pajek dataset [10].
Snap. The last set of instances is a subset of the Stanford large network dataset collection (Snap). It contains 16 graphs derived from an internet topology [71], 9.629 graphs describing user interaction on the music streaming service `deezer` [93], five graphs describing email interactions of members in a large European research institution [72], and ten graphs as friend networks of `Facebook` users [73].

All instances were chosen to have up to 1000 nodes and a possible spreading process application. Some of the instances are directed graphs, some are weighted, others are not connected. If not provided, all edge weights were set to one. For all of the different sources and their different graph formats parsers in `octave` are available. We conducted 100 randomized test runs on each instance (only the 9629 `deezer` graphs were only run once). For each run s was chosen randomly, just as c (exponential distribution with mean 1) and t_s (Gaussian with mean 0 and standard deviation 10). We used $\sigma = \frac{1}{5}c$ as standard deviation for the random error of oracle queries.

5.4 Benchmark Library: Convergence

To evaluate the convergence behavior of Algorithm 2, we assess the quality of the detected source j^* in comparison to the true source s_k of instance k. We use the following normalization and evaluation measure.

Definition 5.4.1 (Normalization)**.** *Let Q be the set of all instances (test runs) and $k \in Q$ a specific one.*

Let i_k be the number of iterations until Algorithm 2 terminated for instance k. We then transform all iterations $i \in \{i_{min} = 3, \dots, i_k\}$ to normalized iterations *i_n via $i_n = (i - i_{min})/(i_k - i_{min}) \in [0, 1]$, omitting the dependence on k for notational simplicity. Then we define*

$$q(k, i_n) := \frac{\left|\left\{j \in V : J^*_{j,\widehat{S}} \leq J^*_{s,\widehat{S}}\right\}\right|}{|V|} \tag{5.1}$$

as the uniqueness level *of a given true source s for an oracle query set $\widehat{S}$. It depends on the instance $k \in Q$ and on the normalized iteration counter i_n of Algorithm 2. The level $q(k, i_n) \in \left[\frac{1}{|V|}, 1\right]$ is*

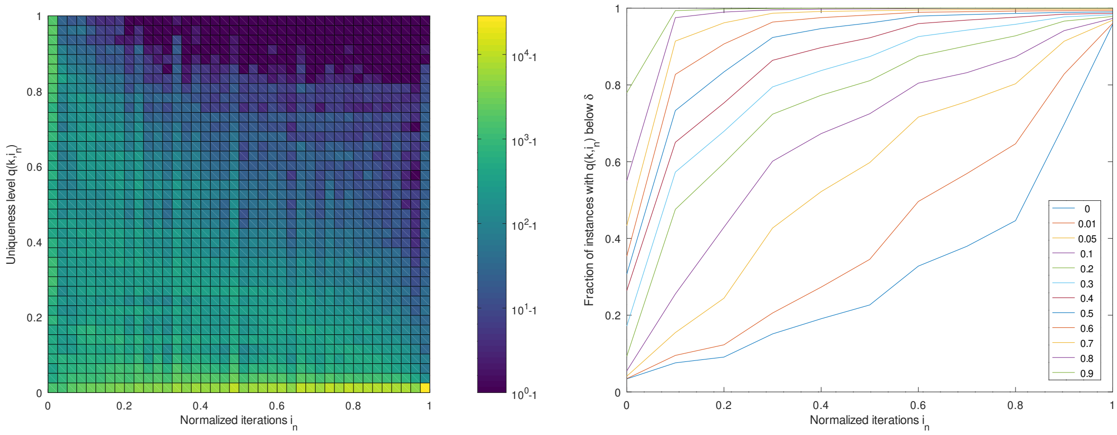


Figure 5.2: Using Definition 5.4.1, $q(k, i_n)$ is visualized for all instances $k \in Q$. Left: a color gradient indicates for how many instances $k \in Q$ the value $q(k, i_n)$ is in a given box. While for small iteration numbers i_n the values $J^*_{s_k,\widehat{S}_{k,i_n}}$ are almost randomly distributed among all $j \in V$, for large iterations we have $J^*_{s_k,\widehat{S}_{k,i_n}} \leq J^*_{j,\widehat{S}_{k,i_n}}$ for almost all instances k and all $j \in V$, indicating the probable proximity of j^* to the true source s_k of instance k. Right: for different values of δ the lines plot the fraction $|Q_1(i_n)|/|Q|$ with $Q_1(i_n) := \{k \in Q : \ q(k, i_n) \leq \delta + \frac{1}{|V|}\}$. E.g., for $\delta = 0$ the lowest blue line depicts the fraction of instances for which at iteration i_n the source s_k was the unique minimizer of J^*, increasing from approximately 5% to 95%.

evaluated for the least squares function $J^*_{j,\widehat{S}_{k,i_n}}$. If $q(k, i_n) = \frac{1}{|V|}$ then $j^* = s$.

First, Table 5.1 shows the median and mean distances between j^* and s after termination (i.e., $i_n = 1$) of Algorithm 2, indicating its accuracy. There are no significant differences between the test sets, indicating the general applicability of Algorithm 2. As the following results are very similar for all test sets, we present them from now on for Q as the union of the test sets Col, Misc, and Snap.

Table 5.1: Distance between j^* at $i_n = 1$ and s for different test sets Q. Note that for test set Misc with weighted graphs the distances of j^* to s were divided by the maximum non-infinite shortest path lengths, and the special infinity treatment was applied, see Remark 5.1.1. Mostly, Algorithm 2 returned $j^* \approx s$.

Problems	Median	Mean	Max	# Inf
Col	0.000	0.0177	3.000	0
Misc	0.000	0.7064	1485.110	12
Misc corrected	0.000	0.0028	0.804	12
Snap	0.000	0.0111	4.000	0

Second, to investigate the efficiency of Algorithm 2, we illustrate in Figure 5.2 the uniqueness level q as a function of normalized iterations and instances $k \in Q$. Both plots indicate that for the chosen sets $\widehat{S}_{k,i_n}$ the termination criterion is a good choice and that Algorithm 2 is well-posed in the sense that at termination, the true source s_k is detected with high probability (as $q(k, i_n = 1) \approx 0$ for almost all $k \in Q$). From Figure 5.2 we deduce on the one hand that an earlier termination of Algorithm 2, as seemed plausible from the example in Subsection 5.2, would often result in j^* that are not minimal with respect to the least squares regression. On the other hand, more iterations are not necessary.

5.5 Benchmark Library: Number of Iterations

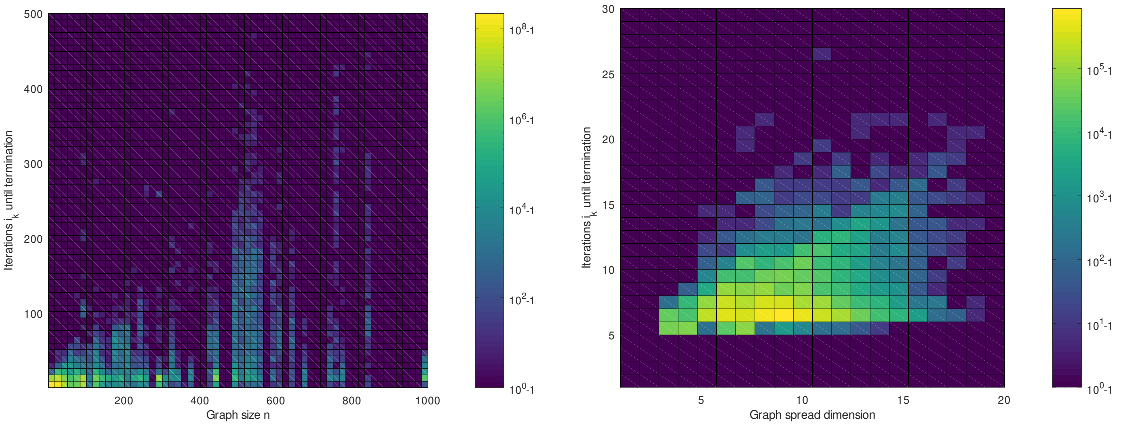


Figure 5.3: Color gradients showing how many instances $k \in Q$ needed (on the y-axis) how many iterations until termination of Algorithm 2. Left: Plotted over the graph size $n = |V|$, suggesting a linear relation $i_k \leq c_1 n$ for a constant c_1 and most $k \in Q$. Right: Plotted over the spread dimension β, suggesting a linear relation $i_k \leq c_2 \beta$ for a constant c_2 and most $k \in Q_{\text{small}}$ where $Q_{\text{small}} \subseteq Q$ contains all graphs in Q with $n \leq 60$. Up to this size a brute force enumeration of the spread dimension was computationally feasible.

In this section we have a closer look at how the iteration numbers i_k until Algorithm 2 terminated relate to properties of the graphs. Figure 5.3 (left) shows them for different graph sizes $n = |V|$. For most $k \in Q$ we have $i_k \leq \frac{1}{2}n$. As the spread dimension is the number of necessary oracle queries (iterations in the online case), this is plausible when looking at the upper bound from Proposition 3.3.11 for complete graphs. Note that the spread dimension is not a strict lower bound on i_k due to the advantage that in the online setting we can place oracle queries with knowledge gained in previous iterations.

For small graphs ($n \leq 60$), we could determine the spread dimension by brute force enumeration. The result in Figure 5.3 (right) confirms the impression that the number of iterations of Algorithm 2 is in many cases below the spread dimension, and only in few cases above. Thus it seems valid to see i_k, at least for the chosen variance of measurement errors, as an approximation of the spread dimension.

This result does not consider other graph properties. An investigation of the topological diameter and of the connectivity (number of edges divided by n) of the graph did not reveal obvious correlations (negative results are not shown here, the color gradients were rather erratic). Known results for the metric dimension β, which is a lower bound for the spread dimension as discussed in Section 3.3.1, indicate that the graph topology could have a strong impact (on the lower and not necessarily active bound). E.g., for the diameter d it was shown that

$$n \leq \left(\left\lfloor \frac{2d}{3} \right\rfloor + 1 \right)^{\beta} + \beta \sum_{i=1}^{\lceil d/3 \rceil} (2i-1)^{\beta-1}$$

by [47, Theorem 3.1]. Also the simpler, but less strict inequality

$$n \leq d^{\beta} + \beta$$

from [59] emphasizes the role of the diameter. Not finding a correlation between the diameter d and i_k might indicate that the diameter is not as relevant for the spread dimension as it is for the metric dimension or that i_k differs from the spread dimension for specific graphs. Note also that the spread dimension depends on the edge weights of the graph. Two graphs with the same edge sets can have different spread dimensions, if the edge weights are different. The connection between graph properties on the one hand and spread dimension and iteration numbers on the other hand should be investigated in future research.

5.6 Benchmark Library: Relaxation of Oracle Query Feasibility

For the previous results the algorithm with Definition 3.4.14 was used, i.e., a maximal difference in the number of oracle queries between different nodes of one was allowed. This can be relaxed to any finite number, not altering the results on conversion of the algorithm in the limit. Short term performance can be impacted by this, as more flexibility for oracle queries is given to the heuristic that chooses which nodes to query. In the Example in Subsection 5.2 it was already noted, that it could have been beneficial not to measure all nodes once, but instead concentrate on nodes revealing most about the source.

Remark 5.6.1 (Convergence with relaxed Oracle Query Feasibility)**.** *The idea to proof that relaxation of Definition 3.4.14 does not alter the convergence result is the following. A maximal finite difference d of oracle queries between different nodes and the graph size n is given. The estimator is agnostic to the order in which the oracle queries are performed. Hence, it is possible to reorder all queries in the following way. The maximal number of queries are put to the front, such that they fulfill Definition 3.4.14, i.e., $d = 1$. All other queries are put at the end, this are at most d times n queries. As in the limit this finite amount of queries does not have influence compared to the infinitely many queries before them, the convergence is the same as before.*

Ten additional randomized test runs were conducted on each instance, each with a different allowed maximal oracle query difference from 1 to 10. Figure 5.3 depicts the results. The left part shows that the distance to the true source of the source estimate has no dependency on the relaxation of feasibility. This is expected, as the termination criteria is independent of Oracle

Query Feasibility. On the right one sees a clear dependency of the iteration number on the Oracle Query Feasibility. This is also expected, as the convergence (and hence the iteration number given a fixed termination criteria) is mainly depended on the oracle queries.

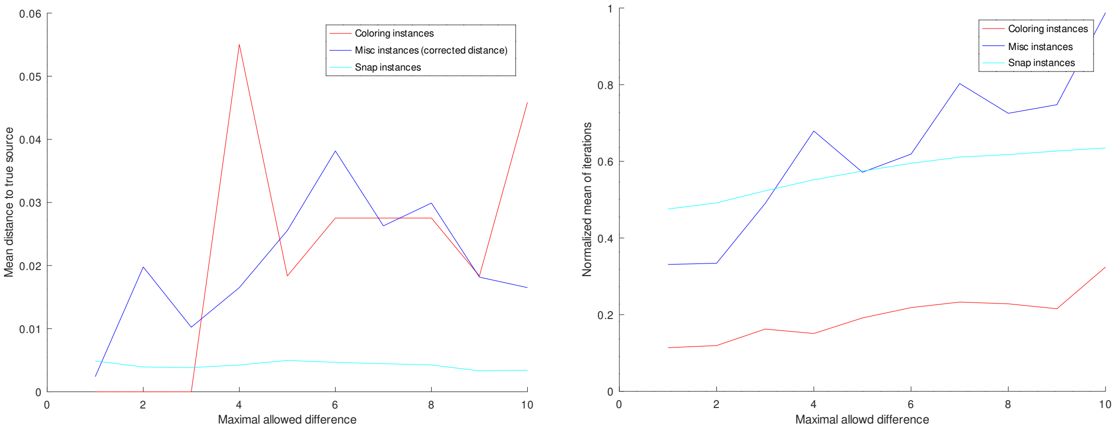


Figure 5.4: Distance to source and iterations are plotted over maximal allowed difference for queries at different nodes. Left: Mean distance to true source for the three test sets, plotted over the allowed maximal oracle query difference from 1 to 10. For the Misc test set corrected values are plotted, i.e., distances divided by the maximum finite shortest path lengths. No trend is visible and values are in general comparable to Table 5.1. Right: Iterations are normalized by problem size. The mean of the normalized iterations for the three test sets is plotted over the allowed maximal oracle query difference from 1 to 10. A clear trend of increasing iterations with feasibility relaxation is visible.

The negative dependency between iteration number and feasibility relaxation could indicate, that the used heuristic is not optimal for the majority of the instances of the test set or that the strong negative trend of some instances dominates the overall mean.

In Figure 5.5 the distribution of slopes of the instances is depicted. There are more negative slopes with larger absolute value explaining the overall negative mean. A small graph instance from the test set Misc with small (large absolute) negative slope is presented in Example 9. As the graph is acyclic, the source estimation is purely combinatorial, cf. Remark 5.1.1. All nodes differ in their infinity pattern, a second query at a node is never required. Source estimation just needs to identify the correct pattern. The problem is that the heuristic behaves deterministically in the following way, if node 8 is the true source: At the start the nodes 1 and 7 are chosen, then nodes 1 to 6 are queried as often as possible (if the maximal difference is 10 then node 1 is queried 9 times), and finally node 8 is queried, leading to a successful source estimate.

The problem in this case is, that the heuristic is based on a continuous variance criterium of linear regression, it does not include direct topology information, previous oracle queries or estimation information.

Example 9 (Directed graph)**.** The graph $G = (V, E)$ has nodes

$$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

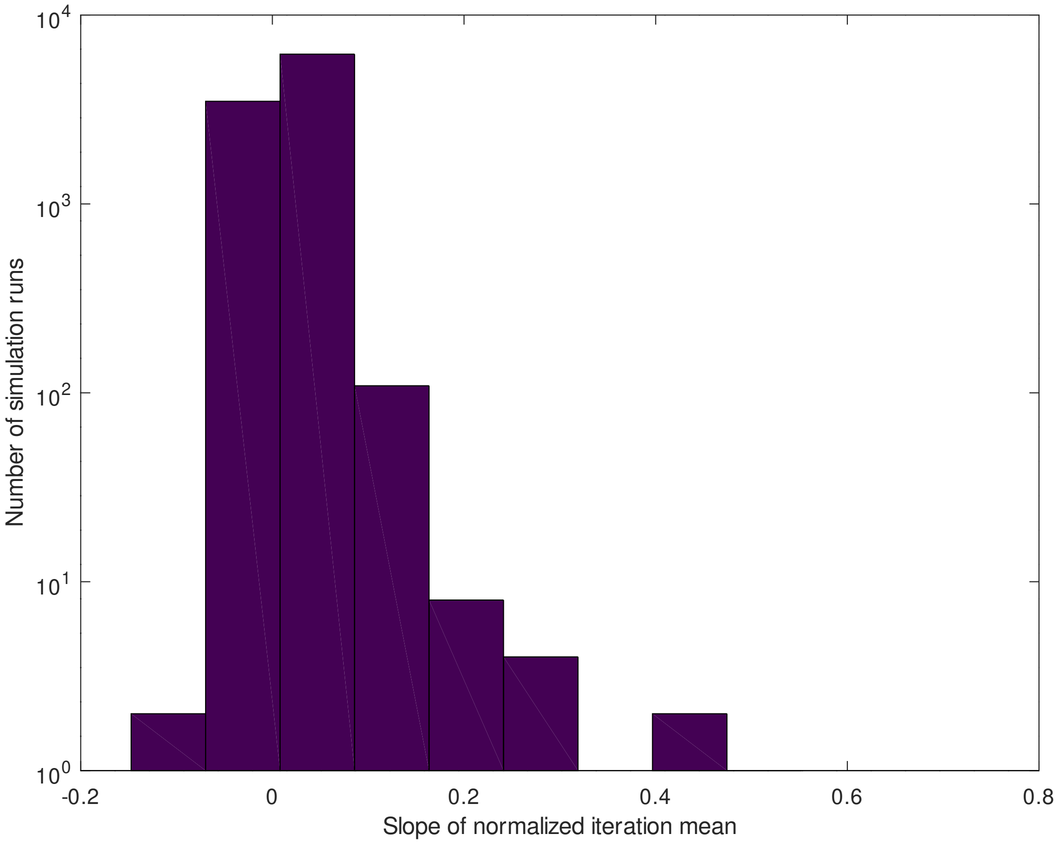


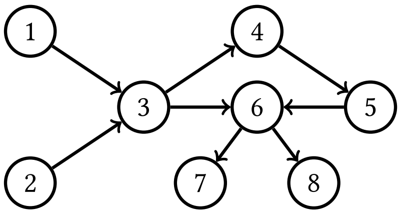
Figure 5.5: For all simulation runs the slope of the normalized iterations over the allowed maximal oracle query difference from 1 to 10 was calculated. This is a histogram of these slopes. The number of runs (y-axis) falling in a certain slope range (x-axis) is indicated by the height of the blue boxes (logarithmic scale).

and directed edges

$$E = \{\{1,3\},\{2,3\},\{3,4\},\{3,6\},\{4,5\},\{5,6\},\{6,7\},\{6,8\}\}.$$

The weights are $\ell(e) = 1$ for all edges except edge $\{5,6\}$ with weight 0.8. The graph has no loops. It is shown in Figure 5.6.

In general the oracle query heuristic can fail, but the Oracle Querry Feasibility, enforces convergence in the long run for every maximal difference value. This gives a lot of freedom to tune the algorithm to the application at hand.



j i	1	2	3	4	5	6	7	8
1	0.0	1.0	∞	2.0	3.0	2.0	3.0	3.0
2	∞	0.0	∞	1.0	2.0	1.0	2.0	2.0
3	∞	1.0	0.0	2.0	3.0	2.0	3.0	3.0
4	∞	∞	∞	0.0	1.0	1.8	2.8	2.8
5	∞	∞	∞	∞	0.0	0.8	1.8	1.8
6	∞	∞	∞	∞	∞	0.0	1.0	1.0
7	∞	∞	∞	∞	∞	∞	0.0	∞
8	∞	∞	∞	∞	∞	∞	∞	0.0

Figure 5.6: Left: visualization of the directed example graph. Right: symmetric matrix with shortest path distances $d_{i,j}$.

6 Conclusion

The thesis introduced an abstract framework for source detection on graphs. First, the source detection problem was defined and a solution for the deterministic offline case was derived based on the concept of spread dimension, which is an extension of the metric dimension. Modular decomposition and split decomposition were applied to efficiently compute the solution and an online algorithm for the deterministic source detection problem using these concepts was proposed.

For stochastic source detection, a general source estimator was introduced and the estimation quality was investigated. In the offline case, it is impossible to know a priori which oracle questions would be sufficient for correct estimation or even for an estimation with a certain probability. Therefore, an online algorithm to overcome this limitation was proposed, which is proven to converge with feasible oracle queries to the true source. Its performance and robustness are demonstrated in extensive numerical simulations.

The algorithm was applied to find the source of cardiac arrhythmias in a medical simulation study, showing promising performance for treatment improvements. In numerical random simulations on problems from a new graph library, the algorithm was robust and effective, with the source estimate usually being correct and the number of iterations (oracle queries) being in the order of the graph size.

Finally, the relaxation of the feasibility in Definition 3.4.14 was investigated in simulations on the same library, with mixed results. The algorithm performance either improved or worsened, depending on the instance, while the general trend was negative, indicating a suboptimal oracle query heuristic with respect to the library. However, even with this suboptimal heuristic, the algorithm generally performed well on the library.

The oracle question placement is currently heuristic and needs to be found depending on the application. A general, theoretically sound, oracle placement strategy that leads to theoretically fast convergence and good source estimates should be investigated. It would also be interesting to know what part of the oracle placement is independent of the estimator and what is needed for a given estimator at hand.

The current linear regression estimator could be improved or extended by various methods from the linear regression literature. Other estimators based on Bayes' theorem or optimal estimators for non-Gaussian error distributions could also be considered, depending on the application. In general, it would be interesting to know what kind of features an estimator would need, to converge under mild assumptions on the oracle questions (e.g., feasibility). The termination criteria should not be heuristic, but should be derived directly from the stochastic error analysis of the used estimator.

All these stochastic algorithm ingredients are only loosely connected to the graph topology, and more research could be done in this area. Up to now, only feasibility is used, and more such criteria may be needed, and the tools should consider the graph topology directly. The algorithm should be applied to more real-world problems, especially larger instances, and the use of decomposition strategies presented in this thesis should be applied.

Another future research direction is to study the problem with more than one source, either in the sense that the signal is received only once everywhere from the nearest source or in the sense that all signals are received, and one does not know which signal is from which source. Another question is whether it is known how many sources are present, or whether this is an unknown problem parameter.

Source detection on graphs is important due to its applications and the relevance of networks in modern life. I believes that these applications will pose theoretical challenges to the theoretical and algorithmic side, inducing fruitful research. The research results will bring forward solutions to practical problems, improving our performance in the management of these networks.

6 Bibliography

[1] J. C. Adams, K. Srivathsan, and W. K. Shen. Advances in management of premature ventricular contractions. *Journal of interventional cardiac electrophysiology*, 35(2):137–149, 2012.

[2] T. Aiba, W. Shimizu, A. Taguchi, K. Suyama, T. Kurita, N. Aihara, and S. Kamakura. Clinical usefulness of a multielectrode basket catheter for idiopathic ventricular tachycardia originating from right ventricular outflow tract. *Journal of Cardiovascular Electrophysiology*, 12(5):511–517, 2001.

[3] S. Albert, J. Pätzold, A. Schiewe, P. Schiewe, and A. Schöbel. Documentation for lintim 2020.02, 2020.

[4] R. Alexander. Solving ordinary differential equations i: Nonstiff problems (e. hairer, sp norsett, and g. wanner). *SIAM Review*, 32(3):485, 1990.

[5] U. Alon. Collection of complex networks.

[6] 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):118701, 2014.

[7] R. H. Anderson, J. Yanni, M. R. Boyett, N. J. Chandler, and H. Dobrzynski. The anatomy of the cardiac conduction system. *Clinical Anatomy: The Official Journal of the American Association of Clinical Anatomists and the British Association of Clinical Anatomists*, 22(1):99–113, 2009.

[8] A. Baker, R. Inverarity, M. Charlton, and S. Richmond. Detecting river pollution using fluorescence spectrophotometry: case studies from the ouseburn, ne england. *Environmental Pollution*, 124(1):57–70, 2003.

[9] F. G. Ball and O. D. Lyne. Optimal vaccination policies for stochastic epidemics among a population of households. *Mathematical biosciences*, 177:333–354, 2002.

[10] V. Batagelj and A. Mrvar. Pajek datasets. `http://vlado.fmf.uni-lj.si/pub/networks/data/`, 2006.

[11] J. E. Beasley. Or-library: Distributing test problems by electronic mail. *The Journal of the Operational Research Society*, 41(11):1069–1072, 1990.

[12] A. Beck, P. Stoica, and J. Li. Exact and approximate solutions of source localization problems. *IEEE Transactions on signal processing*, 56(5):1770–1778, 2008.

[13] V. Belik, T. Geisel, and D. Brockmann. Natural human mobility patterns and spatial spread of infectious diseases. *Physical Review X*, 1(1):011001, 2011.

[14] J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *The Journal of the Acoustical Society of America*, 107(1):384–391, 2000.

[15] A. E. Bernhard and K. G. Field. Identification of nonpoint sources of fecal pollution in coastal waters by using host-specific 16s ribosomal dna genetic markers from fecal anaerobes. *Appl. Environ. Microbiol.*, 66(4):1587–1594, 2000.

[16] J. Bernoulli. Jacobi bernoulli solutio problematum fraternorum. *Acta Eruditorum, Leipzig, May*, 1697:214, 1697.

[17] J. W. Berry, L. Fleischer, W. E. Hart, C. A. Phillips, and J.-P. Watson. Sensor placement in municipal water networks. *Journal of Water Resources Planning and Management*, 131(3):237–243, 2005.

[18] P. Bhagirath, M. van der Graaf, E. van Dongen, J. de Hooge, V. van Driel, H. Ramanna, N. de Groot, and M. J. Götte. Feasibility and accuracy of cardiac magnetic resonance imaging–based whole-heart inverse potential mapping of sinus rhythm and idiopathic ventricular foci. *Journal of the American Heart Association*, 4(10):e002222, 2015.

[19] P. Bianchi, M. Debbah, M. Maïda, and J. Najim. Performance of statistical tests for single-source detection using random matrix theory. *IEEE Transactions on Information theory*, 57(4):2400–2419, 2011.

[20] N. H. Bingham and J. M. Fry. *Regression: Linear models in statistics.* Springer Science & Business Media, 2010.

[21] S. Bornholdt and H. G. Schuster. Handbook of graphs and networks. *From Genome to the Internet, Willey-VCH (2003 Weinheim)*, 2001.

[22] G. Bounova. Octave network toolbox, September 2016.

[23] M. S. Brandstein. A pitch-based approach to time-delay estimation of reverberant speech. In *Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on*, pages 4–pp. IEEE, 1997.

[24] D. Brockmann and D. Helbing. The hidden geometry of complex, network-driven contagion phenomena. *science*, 342(6164):1337–1342, 2013.

[25] J. Cáceres, 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 on discrete mathematics*, 21(2):423–441, 2007.

[26] 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(1-3):99–113, 2000.

[27] G. Chartrand and P. Zhang. The theory and applications of resolvability in graphs: A survey. 160, 01 2003.

[28] J. C. Chen, K. Yao, and R. E. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, 19(2):30–39, 2002.

[29] E. G. Coffman Jr, Z. Ge, V. Misra, and D. Towsley. Network resilience: exploring cascading failures within bgp. In *Proc. 40th Annual Allerton Conference on Communications, Computing and Control*, 2002.

[30] V. Colizza and A. Vespignani. Invasion threshold in heterogeneous metapopulation networks. *Physical review letters*, 99(14):148701, 2007.

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

[32] S. Costanzo, M. O'donohue, W. Dennison, N. Loneragan, and M. Thomas. A new approach for detecting and mapping sewage impacts. *Marine Pollution Bulletin*, 42(2):149–156, 2001.

[33] R. Diestel. *Graph theory*. Springer Publishing Company, Incorporated, 2017.

[34] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring. *GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations*, 2020.

[35] G. Elfving. Optimum allocation in linear regression theory. *The Annals of Mathematical Statistics*, 23(2):255–262, 1952.

[36] D. G. Eliades and M. M. Polycarpou. Fault isolation and impact evaluation of water distribution network contamination. *IFAC Proceedings Volumes*, 44(1):4827–4832, 2011.

[37] V. Fedorov. Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):581–589, 2010.

[38] V. Fioriti and M. Chinnici. Predicting the sources of an outbreak with a spectral technique. *arXiv preprint arXiv:1211.2333*, 2012.

[39] R. A. Fisher. The wave of advance of advantageous genes. *Annals of Human Genetics*, 7(4):355–369, 1937.

[40] C. Fleurent and J. A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63(3):437–461, 1996.

[41] T. Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18(1-2):25–66, 1967.

[42] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1455–1466. IEEE, 2005.

[43] L. Gepstein, G. Hayam, and S. A. Ben-Haim. A novel method for nonfluoroscopic catheter-based electroanatomical mapping of the heart: in vitro and in vivo accuracy results. *Circulation*, 95(6):1611–1622, 1997.

[44] M. Habib, F. De Montgolfier, and C. Paul. A simple linear-time modular decomposition algorithm for graphs, using order extension. In *Scandinavian Workshop on Algorithm Theory*, pages 187–198. Springer, 2004.

[45] S. Hartung and A. Nichterlein. On the parameterized and approximation hardness of metric dimension. In *2013 IEEE Conference on Computational Complexity*, pages 266–276. IEEE, 2013.

[46] M. Hauptmann, R. Schmied, and C. Viehmann. Approximation complexity of metric dimension problem. *Journal of Discrete Algorithms*, 14:214–222, 2012.

[47] C. Hernando, M. Mora, I. M. Pelayo, C. Seara, and D. R. Wood. Extremal graph theory for metric dimension and diameter. *Electronic Notes in Discrete Mathematics*, 29:339–343, 2007.

[48] M. Hocini, A. J. Shah, T. Neumann, M. Kuniss, D. Erkapic, A. Chaumeil, S.-J. COPLEY, P. B. Lim, P. Kanagaratnam, A. Denis, et al. Focal arrhythmia ablation determined by high-resolution noninvasive maps: multicenter feasibility study. *Journal of cardiovascular electrophysiology*, 26(7):754–760, 2015.

[49] A. M. Hopkins, C. Miller, A. Connolly, C. Genovese, R. C. Nichol, and L. Wasserman. A new source detection algorithm using the false-discovery rate. *The Astronomical Journal*, 123(2):1086, 2002.

[50] P. A. Iaizzo. *Handbook of cardiac anatomy, physiology, and devices.* Springer Science & Business Media, 2009.

[51] M. A. Jatoi, N. Kamel, A. S. Malik, I. Faye, and T. Begum. A survey of methods used for source localization using eeg signals. *Biomedical Signal Processing and Control*, 11:42–52, 2014.

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

[53] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part ii, graph coloring and number partitioning. *Operations research*, 39(3):378–406, 1991.

[54] D. S. Johnson and M. A. Trick. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge, October 11-13, 1993*, volume 26. American Mathematical Soc., 1996.

[55] B. Joseph-Duran, M. N. Jung, C. Ocampo-Martinez, S. Sager, and G. Cembrano. Minimization of sewage network overflow. *Water Resources Management*, 28(1):41–63, 2014.

[56] H. Kanamori and L. Rivera. Source inversion of w phase: speeding up seismic tsunami warning. *Geophysical Journal International*, 175(1):222–238, 2008.

[57] A. Kazemi and A. Mohamed. An new approach for location voltage sag source in a power system by using regression coefficients. In *Regional Engineering Postgraduate Conference (EPC)*, pages 374–381, 2010.

[58] A. Kazemi, A. Mohamed, H. Shareef, and H. Zayandehroodi. Review of voltage sag source identification methods for power quality diagnosis. *Przegląd Elektrotechniczny*, 89(8):143–146, 2013.

[59] S. Khuller, B. Raghavachari, and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70(3):217–229, 1996.

[60] J. Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 272–319, 1959.

[61] J. Kiefer and J. Wolfowitz. Optimum designs in regression problems. *The Annals of Mathematical Statistics*, pages 271–294, 1959.

[62] D. E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. AcM Press New York, 1993.

[63] A. Kolmogorov, I. Petrovskii, and N. Piskunov. A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem. *Selected Works of AN Kolmogorov I*, pages 248–270, 1937.

[64] D. König. *Theorie der endlichen und unendlichen Graphen: Kombinatorische Topologie der Streckenkomplexe*, volume 16. Akademische Verlagsgesellschaft mbh, 1936.

[65] J. Kratica, M. Čangalović, and V. Kovačević-Vujčić. Computing minimal doubly resolving sets of graphs. *Computers & Operations Research*, 36(7):2149–2159, 2009.

[66] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine*, 13(4):67–94, 1996.

[67] C. Laird, L. Biegler, B. van Bloemen Waanders, and R. Bartlett. Time dependent contamination source determination for municipal water networks using large scale optimization. *Journal of Water Resources Planning and Management*, 2003.

[68] A. Latheef, M. Negnevitsky, and V. Faybisovich. Voltage sag source location identification. In *CIRED 2009-20th International Conference and Exhibition on Electricity Distribution-Part 1*, pages 1–4. IET, 2009.

[69] R. C. Leborgne and D. Karlsson. Voltage sag source location based on voltage measurements only. *Electrical Power Quality and Utilisation. Journal*, 14:25–30, 2008.

[70] F. T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979.

[71] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, 2005.

[72] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[73] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[74] W. Luo, W. P. Tay, and M. Leng. How to identify an infection source with limited observations. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):586–597, 2014.

[75] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[76] D. Malioutov, M. Cetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE transactions on signal processing*, 53(8):3010–3022, 2005.

[77] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[78] M. S. Marlim and D. Kang. Identifying contaminant intrusion in water distribution networks under water flow and sensor report time uncertainties. *Water*, 12(11):3179, 2020.

[79] R. M. McConnell and F. De Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):198–209, 2005.

[80] R. A. Melter and I. Tomescu. Metric bases in digital geometry. *Computer Vision, Graphics, and Image Processing*, 25(1):113–121, 1984.

[81] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.

[82] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[83] R. H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In *Graphs and Order*, pages 41–101. Springer, 1985.

[84] R. H. Möhring and F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. In *North-Holland mathematics studies*, volume 95, pages 257–355. Elsevier, 1984.

[85] M. E. Newman. Network data. `http://www-personal.umich.edu/~mejn/netdata/`.

[86] M. E. Newman. Spread of epidemic disease on networks. *Physical review E*, 66(1):116–128, 2002.

[87] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[88] J. A. Noblet, D. L. Young, E. Y. Zeng, and S. Ensari. Use of fecal steroids to infer the sources of fecal indicator bacteria in the lower santa ana river watershed, california: sewage is unlikely a significant source. *Environmental science & technology*, 38(22):6002–6008, 2004.

[89] A. K. Pradhan and A. Routray. Applying distance relay for voltage sag source detection. *IEEE Transactions on Power Delivery*, 20(1):529–531, 2005.

[90] S. G. Priori, C. Blomström-Lundqvist, A. Mazzanti, N. Blom, M. Borggrefe, J. Camm, P. M. Elliott, D. Fitzsimons, R. Hatala, et al. 2015 esc guidelines for the management of patients with ventricular arrhythmias and the prevention of sudden cardiac death: The task force for the management of patients with ventricular arrhythmias and the prevention of sudden cardiac death of the european society of cardiology (esc) endorsed by: Association for european paediatric and congenital cardiology (aepc). *Ep Europace*, 17(11):1601–1687, 2015.

[91] T. Randell. Medical and legal considerations of brain death. *Acta anaesthesiologica scandinavica*, 48(2):139–144, 2004.

[92] L. A. Rossman et al. Epanet 2: users manual. 2000.

[93] B. Rozemberczki, O. Kiss, and R. Sarkar. An api oriented open-source python framework for unsupervised learning on graphs, 2020.

[94] A. Salamon. Modular decomposition of directed graphs. CPAN, 2004.

[95] C. Schmitt, G. Ndrepepa, S. Weber, S. Schmieder, S. Weyerbrock, M. Schneider, M. R. Karch, I. Deisenhofer, J. Schreieck, B. Zrenner, et al. Biatrial multisite mapping of atrial premature complexes triggering onset of atrial fibrillation. *The American journal of cardiology*, 89(12):1381–1387, 2002.

[96] D. Shah and T. Zaman. Detecting sources of computer viruses in networks: Theory and experiment. *SIGMETRICS Perform. Eval. Rev.*, 38(1):203–214, June 2010.

[97] X. Sheng and Y.-H. Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1):44–53, 2005.

[98] J. Sidhu, W. Ahmed, W. Gernjak, R. Aryal, D. McCarthy, A. Palmer, P. Kolotelo, and S. Toze. Sewage pollution in urban stormwater runoff as evident from the widespread presence of multiple microbial and chemical source tracking markers. *Science of the Total Environment*, 463:488–496, 2013.

[99] S. Singh, M. Ordaz, J. Pacheco, and F. Courboulex. A simple source inversion scheme for displacement seismograms recorded at short distances. *Journal of seismology*, 4(3):267–284, 2000.

[100] L. Slotta-Bachmayr. How burial time of avalanche victims is influenced by rescue method: An analysis of search reports from the alps. *Natural Hazards*, 34(3):341–352, 2005.

[101] K. Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 12(1/2):1–85, 1918.

[102] R. C. Tillquist, R. M. Frongillo, and M. E. Lladser. Getting the lay of the land in discrete space: A survey of metric dimension and its applications. *arXiv preprint arXiv:2104.07201*, 2021.

[103] A. Wald. On the efficient design of statistical investigations. *The Annals of Mathematical Statistics*, 14(2):134–140, 1943.

[104] H. Wang and P. Chu. Voice source localization for automatic camera pointing system in videoconferencing. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1, pages 187–190. IEEE, 1997.

[105] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[106] T. Weber, V. Kaibel, and S. Sager. Source detection on graphs. Pre-print available `https://optimization-online.org/?p=19126`.

[107] T. Weber, H. A. Katus, S. Sager, and E. P. Scholz. Novel algorithm for accelerated electroanatomic mapping and prediction of earliest activation of focal cardiac arrhythmias using mathematical optimization. *Heart rhythm*, 14(6):875–882, 2017.

[108] S. Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[109] K. Yao, R. E. Hudson, C. W. Reed, D. Chen, and F. Lorenzelli. Blind beamforming on a randomly distributed sensor array system. *IEEE Journal on Selected Areas in Communications*, 16(8):1555–1567, 1998.

[110] P.-D. Yu, C. W. Tan, and H.-L. Fu. Epidemic source detection in contact tracing networks: Epidemic centrality in graphs and message-passing algorithms. *arXiv preprint arXiv:2201.06751*, 2022.

[111] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[112] M. Zang, T. Zhang, J. Mao, S. Zhou, and B. He. Beneficial effects of catheter ablation of frequent premature ventricular complexes on left ventricular function. *Heart*, 100(10):787–793, 2014.

[113] F. Zhang, B. Yang, H. Chen, W. Ju, P. Kojodjojo, M. Li, K. Gu, G. Yang, K. Cao, and M. Chen. Non-contact mapping-guided ablation of ventricular arrhythmias originating from the pulmonary artery. *EP Europace*, 18(2):281–287, 2016.