Spring 2018

A network tomography approach for traffic monitoring in smart cities

Ruoxi Zhang

Department:

A NETWORK TOMOGRAPHY APPROACH FOR TRAFFIC MONITORING IN SMART CITIES

by

RUOXI ZHANG

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

COMPUTER SCIENCE

2018

Approved by

Simone Silvestri, Advisor
Sanjay Madria
Bruce McMillin

PUBLICATION THESIS OPTION

This thesis consists of the following article which has been submitted for publication:

Paper I: Pages 3-35 have been accepted by IEEE Transactions on Intelligent Transportation System

ABSTRACT

Various urban planning and managing activities required by a Smart City are feasible because of traffic monitoring. As such, the thesis proposes a network tomography-based approach that can be applied to road networks to achieve a cost-efficient, flexible, and scalable monitor deployment. Due to the algebraic approach of network tomography, the selection of monitoring intersections can be solved through the use of matrices, with its rows representing paths between two intersections, and its columns representing links in the road network. Because the goal of the algorithm is to provide a cost-efficient, minimum error, and high coverage monitor set, this problem can be translated into an optimization problem over a matroid, which can be solved efficiently by a greedy algorithm. Also as supplementary, the approach is capable of handling noisy measurements and a measurement-to-path matching. The approach proves a low error and a 90% coverage with only 20% nodes selected as monitors in a downtown San Francisco, CA topology.

ACKNOWLEDGMENTS

Infinite thanks to Dr. Simone Silvestri for his training, motivation, and support. His assistance in helping me developing my skills and knowledge in the professional field of computer science is sincerely appreciated. I am incredibly grateful to have him as my advisor. Thanks for the opportunity of research that he kindly offered when I had no experience but with sincere passion.

Special thanks also to Dr. Marco Ortolani for his contribution. All discussions with Dr. Ortolani had been meaningful to me. I learned problem-solving skills from him.

I would like to express my profound thanks to Dr. Sanjay Madria and Dr. Bruce McMillin for serving on my committee and Dr. Wei Jiang for his advice on my research. Also, I would like to express my gratitude to all staffs in the computer science department for their help during my time at Missouri S&T.

Finally, I would like to say a word of thanks to my mother, Lijuan Hou, for her love and support.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION

1. INTRODUCTION

Road traffic monitoring is a fundamental component of future Smart Cities. Urban planners need dynamic traffic data to control traffic flow and construct road network. Effective and efficient traffic monitoring approaches are in need to inquire information such as the average traveling time from one location to another. Current traffic monitoring commonly relies on crowdsourcing or deploying sensors. The thesis presents a network tomography based camera deployment strategy while considering cost, coverage, and accuracy.

Scientists have investigated network tomography as a useful tool to estimate internal network behavior, such as delay distribution and packet loss rate [1]. It is suitable for large and diverse networks. In particular, a subset of nodes that located at the edge of the network is selected as monitors. Monitors exchange probing packets between each other and collect end-to-end measurements along paths. Within each time window, a control center collects these measurements centrally, and linear algebraic methods are applied to infer link-level delay. If solves the linear system infers the delay of an individual link successfully, the link is identifiable. Link identifiability depends on network topology, selected monitors, gathered measurements.

To adopt network tomography from communication networks to road networks, we use nodes to present road intersections and edges for roads. Selected intersections are equipped with cameras to identify vehicles and save its pass-by timing. An end-to-end measurement is complete and recorded once the same vehicle reaches another monitoring location. The delay is the average traveling time along that path. Assume the cost of

selecting a node as a monitor varies by locations. The goal of the research is to maximize identifiability while minimizing the deployment cost and inference error. The following paper addressed in details an algorithm to solve the optimization problem.

Network tomography is a promising approach for road traffic monitoring with challenges. For example, the difference in driving speed leads to noise in measurements. The challenge is formulated as a linear optimization problem. Finding a minimum margin of noise and solving linear inequalities minimize the error due to noisy inputs. Also, there are uncertainty and unknown in determining the driving path of a vehicle since it is uncontrollable by the probing system. An algorithm assigning measurements to paths is designed inspired by Kernighan- Lin graph partitioning algorithm [3].

To evaluate the performance of designed algorithms, experiments are applied to both downtown San Francisco networks and synthetic QuadTree-based networks. The solutions retrieved by the proposed network tomography approach are compared to the results of weighted vertex cover [2] and k-means.

PAPER

I. A NETWORK TOMOGRAPHY APPROACH FOR TRAFFIC MONITORING IN SMART CITIES

Ruoxi Zhang Co-Author

Department of Computer Science

Missouri University of Science and Technology

Rolla, Missouri 65401

Tel: 417–298–5267

Email: rz9p4@mst.edu

Sara Newman Co-Author

Department of Computer Science

Missouri University of Science and Technology

Email: sn976@mst.edu

Marco Ortolani Co-Author

Department of Computer Science

University of Palermo, Italy

Email: marco.ortolani@unipa.it

Simone Silvestri Co-Author

Department of Computer Science

University of Kentucky

Lexington, Kentucky 40506-0633

Tel: 859–323–7276

Email: silvestri@cs.uky.edu

ABSTRACT

Traffic monitoring is a crucial enabler for several planning and management activities of a Smart City. However, traditional techniques are often not cost efficient, flexible, and scalable. This paper proposes an approach to traffic monitoring that does not rely on probe vehicles, nor requires vehicle localization through GPS. Conversely, it exploits just a limited number of cameras placed at road intersections to measure car end-to-end traveling times. We model the problem within the theoretical framework of network tomography, to infer the traveling times of all individual road segments in the road network. We specifically deal with the potential presence of noisy measurements, and the unpredictability of vehicles paths. Moreover, we address the issue of optimally placing the monitoring cameras in order to maximize coverage, while minimizing the inference error, and the overall cost. We provide an extensive experimental assessment on the topology of downtown San Francisco, CA, using real measurements obtained through the Google Maps APIs, as well as on realistic synthetic networks. Our approach provides a very low error in estimating the traveling times of more than 95% of all roads even when as few as 20% of road intersections are equipped with cameras.

Keywords: traffic monitoring, network tomography, smart cities

1. INTRODUCTION

The Smart City paradigm is continuously gaining momentum, also thanks to innovative applications for the efficient management of a city's assets leveraged by the availability of pervasive monitoring devices. According to the Smart City Council [32], efficient traffic monitoring is a key enabler to improve urban livability, and sustainability by optimizing traffic flow, road construction, and urban planning in general [3].

Collecting continuous and detailed traffic information is challenging. Previous work in this area has often relied on the deployment of traffic sensors, such as inductive loop detectors embedded in pavements [30], or cameras placed at strategic spots [7], but those

approaches incur in high deployment and maintenance cost, and can only provide local information. Other strategies have been proposed that make use of probe vehicles, such as taxis, buses, and Unmanned Aerial Vehicles [8, 9, 16]; however, probe vehicles do not represent the actual traffic flow and are not pervasively available in many cities. Crowd sensing technologies, such as smartphones, have also been proposed for the evaluation of urban dynamics [33]; although potentially effective, these approaches may pose a threat for the user's privacy and can be the target of cyber attacks [1].

An interesting alternative approach that has been discussed in the literature is based on Network Tomography, an efficient theoretical tool to estimate the internal state of a network by relying only on end-to-end measurements [4]. This framework has been traditionally investigated in the context of computer networks, but it has also been applied to other contexts, including vehicular traffic estimation [27, 31]. A limited number of monitors are coupled to nodes at the edge of the network so that they can exchange probing packets to collect end-to-end measurements of the network paths. The end-to-end measurements are related to the unknown individual link measurements by solving a linear system of equations. Clearly, monitors need to be strategically placed in order to maximize the number of identifiable links, i.e., the number of links for which a unique solution exists.

In this paper, we use network tomography to devise a novel approach for vehicular traffic monitoring in Smart Cities. We assume that cameras are statically placed at few selected road intersections so that they will play the role of monitors in the network tomography terminology. Similarly, cars traversing the roads between pairs of cameras represent the probing packets. At each of the monitored intersections, images of car license plates are captured and sent to a Centralized Traffic Control Center (TCC) for analysis, so that the corresponding end-to-end traveling times may be inferred. The TCC constructs and solves an optimization problem to infer the traffic conditions on all individual roads segments for which only end-to-end traveling times had initially been collected.

The application of Network Tomography to vehicular traffic is not straightforward, and several challenges need to be addressed. Specifically, in communication networks probe packets are assumed to follow a predefined (possibly source-assigned) path [4]. Predefined paths apparently do not hold for vehicles, which are entirely independent, and may very well follow multiple paths between any pair of cameras. Besides, end-to-end traveling times are affected by non-negligible noise, which might make the resulting linear system impossible.

To address those issues, we provide theoretical proofs that allow us to formulate an optimization problem for the camera placement problem to obtain maximum coverage and minimum error while minimizing the cost of deploying cameras. Furthermore, we formulate a linear optimization problem to minimize the margin of error for measurements, so that it is possible to solve the linear system. Finally, we design a greedy approach inspired by the Kernighan-Lin graph partitioning algorithm [17] to assign end-to-end measurements to paths between cameras.

We test our approach on real network topologies of the downtown San Francisco area and generate realistic traveling times for cars over such topologies by a script based on the Google Maps API [12]. Our results show that our approach can cover more than 95% of the road network when as few as 20% of road intersections are equipped with cameras.

2. RELATED WORK

Traditional approaches to urban traffic monitoring rely on the city infrastructure to gather information about road conditions and usage; for instance, data related to traffic flow is often directly measured through inductive loop detectors embedded in pavements [30], or inferred by cameras placed at strategic spots, such as road intersections [7]. Even though extensive deployment may be devised, such approaches are not very cost-effective, and can hardly obtain pervasive coverage.

Other strategies have been proposed that require some form of collaboration on part of the involved entities, either through devices installed on board of commercial or public transportation vehicles, such as taxis or buses; the authors of [8], for instance, describe a system for real-time estimate of the average speed of traffic through GPS sensors installed in the public transportation fleet of a Greek city. Direct control over the monitoring equipment clearly allows for more flexibility; however bus routes, for instance, are not necessarily representative of the general traffic conditions due to dedicated routes, frequent stops, and so on.

Recently, the use of a more advanced technology has been proposed; a system employing Vehicular Sensor Networks for traffic estimation and optimization has been proposed in [9]. In this approach, probing vehicles that equipped with on-board units play the role of mobile sensors. They traverse the city and sense traffic condition to form centralized traffic matrix. By planning the path for some probing vehicles according to sample-error correlation and sampling rules, the uneven sampling issue caused by using public transportation vehicles is solved. Afterward, to reconstruct un-sampled data, a matrix completion based estimation method can be applied to the uniform randomly sampled traffic matrix.

In addition, another advanced technology, the use of Unmanned Aerial Vehicles (UAVs) for real-time traffic monitoring and management is surveyed in [16]. UAVs serves a border range of traffic monitoring requirements, which is not limited to traffic conditions. It can react in real-time to emergency and congestion. The inherent drawback of such approaches is that an extensive investment needs to be made, which can be prohibitively expensive and might make testing impractical.

Finally, an alternative line of research has suggested the use of crowd sensing technologies for the evaluation of urban dynamics. For example, using smartphones as probes to perform traffic detection [33]. In this particular approach, traffic monitoring also relies on existed mobile cellular network infrastructure. Both entering cell and exiting

cell reveal location and timing information. As limitations, spatial accuracy and time-lag problem need to be concerned for such system. Also, crowd sensing, in general, impose extra costs to users, necessarily require user-provided information, which can be manipulated, and may also pose a threat with regards to users' privacy.

Our goals set our approach apart from those proposals. First of all, no explicit collaboration from the vehicles is assumed, that merely act as "passive" probes, and we do not require cars to be equipped with tracking devices, nor we assume any explicit collaborations from the users. We instead chose to rely on the theoretical grounds provided by network tomography [13, 19, 26, 34]. The authors of [13] studied network tomography to identify constant and additive link metrics, and it can be done on any network with three-edge connectivity by using a single monitor and its monitoring cycles. A probing nodes selection algorithm under dynamic routing is presented in [19]. The algorithm computes the monitor-able edge set for each candidate node under all possible routing protocols, and then it discards some candidate nodes to form a subset of remaining probing nodes, which can still obtain a full edge coverage in a given network. Instead of probing nodes selection, [34] addressed a path selection algorithm, and a set of paths is evaluated according to the expected rank of the linear system in the presence of network failures.

Traditional approaches in this field have been proven successful in reconstructing the set of all end-to-end measurements by probing a basis of paths determined by rank decomposition techniques [36]. Nevertheless, these approaches are designed for internet traffic on communication networks, and probe packets are assumed to be source routed. This assumption does not hold in our context, where drivers are free to choose their path.

Only a few papers consider network tomography applied to vehicular traffic monitoring, but these papers do not deal with the placement of cameras nor the presence of noisy measurements [21, 31]. For example, [31] compared the monitoring quality of individuals sensor with the performance of multi-sensor cooperated. In that case, network tomography is applied to data given by sensors located at Seattle area to produce traffic flow informa-

tion. [21] reviewed network tomography related applications and developments. The survey covers both active routing and passive routing, along with two standard approaches, node oriented and path oriented. Traffic monitoring is addressed as a particular application for passive routing in the paper, but it concentrates on the routing matrix estimation problem for solving linear equalities.

In [26], a linear programming formulation is proposed in the context of network tomography to maximize the load carried on the network. This formulation differs from the objectives and constraints of the optimization problems studied in this paper, and therefore cannot be directly compared.

Artificial Intelligence approaches to infer individual road segment delays given some, possibly noisy, traveling times obtained from cameras have been proposed in [14, 15, 27]. These approaches are based on kernel ridge regression [15] and inverse Markov chains [14, 27]. However, these solutions do not address the placement of cameras at intersections to obtain the end-to-end delays, and therefore cannot be directly compared to ours.

Overall, our framework advances previously proposed approaches by comprehensively solving the challenges that arise by applying network tomography to vehicular networks, to provide a cost-efficient, flexible and accurate way for traffic monitoring in Smart Cities.

3. BACKGROUND ON NETWORK TOMOGRAPHY

Network tomography was originally designed for communication networks, such as the Internet, to infer the internal network state through end-to-end measurements taken by monitors, typically strategically placed at the edge of the network [24]. A network is modeled as an undirected graph $G = (V, E)$, where V is the set of nodes, and E is the set of links. Some of the nodes are selected as monitors and will exchange probing packets in order collect end-to-end measurements of the metric of interest (e.g. the overall delay, or packet loss, across the path between two monitors). With reference to the toy example shown

in Figure 1, monitor nodes are indicated by shadowed circles, and thick lines highlight three of the possible paths between any pair of such monitors, namely $p_1(m_1, m_3) = \{l_1, l_7\}$, $p_2(m_1, m_9) = \{l_1, l_{15}\}$, and $p_3(m_9, m_3) = \{l_{15}, l_7\}$.

The fundamental assumption is the additive nature of the metric to be inferred. As an example, the delay of an end-to-end path is the sum of the delays of its links. If we let b_i indicate the measurement obtained by probing path i, and by x_j the (yet unknown) delay on link j^{th}, measurements collected on the path between monitors m_1 and m_9 in Figure 1 would be expressed as: $x_1 + x_{15} = b_1$.

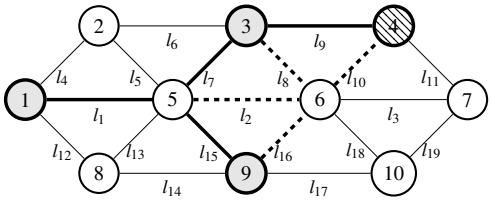
In general, let $P = \{p_1, p_2, ..., p_{|P|}\}$ represent the set of probing paths between monitors. The relation between paths and links is represented by a binary matrix R, of size $|P| \times |E|$, where each row refers to a specific probing path. More specifically, element (i, j) of matrix R is set to 1 if link l_j belongs to path p_i, and to 0 otherwise. The end-to-end measurements are stored in a $1 \times |P|$ vector $\mathbf{b}$, whose elements b_i represent the end-to-end measurements over path p_i.

Hence, if we let x_j represent the delay along link l_j, then the overall delay on path p_i may be expressed as $\sum_{j=1}^{|E|} r_{ij} x_j = b_i$. We can easily extend this over all paths, and formulate the following linear system:

$$R\mathbf{x} = \mathbf{b} \tag{1}$$

where $\mathbf{x}$ represents the individual link measurements. By solving the linear system, the value of x is inferred.

In Figure 1, nodes 1 and 9 are selected as monitors. Adding node 3, and solving the corresponding linear system would completely determine the traveling times on all involved links which are thus said to be identifiable. Only a limited set of monitors was used. Node 5, in particular, was not selected as a monitor, so no camera would need to be deployed at the corresponding road intersection. It is worth noting that a different choice of monitors, covering a larger part of the network, however, does not necessarily improve identifiability; for instance, with monitors $\{1, 4, 9\}$ and paths $p_1(m_1, m_4) = \{l_1, l_7, l_9\}$, $p_2(m_1, m_9) = \{l_1, l_{15}\}$,



$$\begin{cases} l_1 + l_7 + l_9 & = b_1 \\ l_1 + l_2 + l_8 + l_9 & = b_2 \\ l_1 + l_{15} & = b_3 \\ l_{15} + l_7 + l_9 & = b_4 \\ l_{16} + l_8 + l_9 & = b_5 \end{cases}$$

$$R = \begin{bmatrix} 1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0 \end{bmatrix}$$

Figure 1. A sample network with 10 nodes and 19 links; the linear system for the choice of nodes $\{1, 4, 9\}$ as monitors, and multiple possible paths, and the corresponding matrix.

and $p_3(m_9, m_4) = \{l_{15}, l_7, l_9\}$, the system cannot produce a unique solution for links l_7 and l_9. Considering additional paths would not guarantee an improvement in identifiability either. Several works focused on monitor placement and path selection to maximize identifiability in communication networks [24, 34]. However, applying network tomography to vehicular networks introduces novel and unique challenges.

4. A FRAMEWORK FOR VEHICULAR TRAFFIC MONITORING BASED ON NETWORK TOMOGRAPHY

The goal of our framework is to exploit network tomography to provide an accurate estimation of the average traveling time for each road segment in a road network relying only on end-to-end measurements detected by cameras placed at road intersections. To this purpose, we begin by stating our model and assumptions, and then describe the three main modules of our framework. Table 1 summarizes the relevant notation used in the paper.

Table 1. Summary of relevant notation.

Symbol	Description
$G = (V, E)$	Road network G, E road segments and V intersections.
$V' \subseteq V$	Set of candidate intersections for camera placement.
$M \subseteq V'$	Set of selected intersections for camera placement.
c_m	Cost of placing a camera at intersection $m \in V$.
$l \in E$	Road segment, or link.
x_l, $\mathbf{x}$	Traveling time for link l, vector of link traveling times.
b_i, $\mathbf{b}$	Traveling time for path p_i, vector of path traveling times.
P	Set of paths between the monitors in V'.
R	Path matrix, $R[i, j] = 1$ if l_j belongs to p_i, 0 otherwise.
$P_{s,d}$	Set of paths between cameras $m_s, m_d \in M$
$T_{s,d}$	Set of measurements collected by cameras $m_s, m_d \in M$
T_i	Set of measurements assigned to path $p_i \in P_{s,d}$

4.1. MODEL AND ASSUMPTIONS. Given the map of the Smart City's road network, we create a graph $G = (V, E)$ where V is the set of nodes representing road intersections, and E the set of road segments connecting them. In the following we use the term "road intersection" and "node" interchangeably; similarly, we use "road segment" and "link" as synonyms. With no loss of generality, we also assume that all links are symmetrical. Each road segment $l \in E$ is characterized by an average traveling time x_l, which represents the time for a car to traverse the link l, averaged over several measurements in order to account for variability in travel speed.

Our only requirement in terms of hardware is that a limited number of cameras are deployed at road intersections to collect the images of license plates belonging to cars going through that intersection. We do not concern ourselves here specifically with the issue of automatic detection of license plates, as this is a well-investigated problem in image analysis and can be performed with very high accuracy [10].

We assume that cameras can be deployed at a subset $V' \subseteq V$ of all possible intersections. For each intersection $m \in V'$ the cost of placing a camera is c_m. The actual set $M \subseteq V'$ of intersections where cameras are placed is determined by the approach discussed in Section 4.2. If we consider a pair of cameras deployed at intersections $m_1, m_2 \in M$, as a vehicle passes through m_1 and m_2 the corresponding time stamps are sent to the TCC which can thus compute the traveling time for that specific vehicle.

4.2. CAMERA PLACEMENT PROBLEM. The first problem we address is the selection of the set M of intersections where cameras need to be deployed. Ideally, our choice of nodes should incur minimum cost, while providing maximum coverage and a minimum estimation error.

Let us consider the linear system $R\mathbf{x} = \mathbf{b}$ resulting by placing cameras at all possible intersections in V'. Obviously, this solution would provide maximum coverage and minimum error, however, it would incur a very high cost. By using network tomography, we are able to use a much smaller set $M \subseteq V'$ and provide both same coverage and error, for a significantly lower cost. To this purpose, we recall the concept of *basis* of the matrix R.

Definition 1 (Basis [22])**.** *Given a binary matrix R of size $|P| \times |E|$, a basis B is a maximal subset of linearly independent rows (paths).*

In the following theorems, we show that any basis of the matrix R has two relevant properties, namely, it provides both maximum coverage and minimum error. In the following, we use path matrices such as R as sets of vector, to ease the notation.

Theorem 2. *Given a binary matrix R of size $|P| \times |E|$, for any basis B of R, if there exists a path in P that covers a link $l \in E$, then there exists a path in B that covers l.*

Proof. We prove the statement by contradiction. Let us consider a basis $B = \{v_1, \ldots, v_n\}$, and assume that there exists a link l that is not covered by any vector in B. However, since l is covered in R, then there must exist at least a vector $\hat{v} \in R$ that covers l and, since B is a

basis, it should be possible to express $\hat{v}$ as a linear combination of $\{v_1, \ldots, v_n\}$. However, we assumed that l is not covered in B, i.e., $v_i[l] = 0$ for each $i = 1, \ldots, n$. Therefore, $\hat{v}$ cannot be expressed as a linear combination of the paths in B, thus B is not a basis of R, which leads to a contradiction. □

We now focus on the properties of a basis relative to the inference error. By solving the linear system $R\mathbf{x} = \mathbf{b}$, some links are identifiable. Therefore the error for those links is zero, provided that the measurements in $\mathbf{b}$ are accurate. However, for links that are not identifiable, multiple values exist that would satisfy the system. Our algorithm for inferring values for unidentifiable links is described in detail in Section 4.3, but intuitively the larger the space of possible choices, the larger the potential inference error for the unidentifiable links.

More formally, let $Q \subseteq R$ be any set of paths whose end points we are planning to monitor through cameras. Furthermore, let $Vol(Q)$ be the volume of the polyhedron of feasible solutions for the linear system considering only the equations in $R\mathbf{x} = \mathbf{b}$ corresponding to the paths in Q. We denote such smaller system as $Q\mathbf{x}_Q = \mathbf{b}_Q$. The larger $Vol(Q)$, the higher the error we may cause by picking a point in the polyhedron, conversely if $Vol(Q)$ is smaller the inference error is also smaller[1].

Considering all paths in R obviously allows us to obtain the minimum number of unidentifiable links, as well as the minimum volume of $Vol(R)$. It is well known in linear algebra that if we restrict ourselves to consider a basis B of R, and solve the corresponding linear system $B\mathbf{x}_B = \mathbf{b}_B$, we obtain the same set of unidentifiable links [5]. In the following theorem, we further show that such system implies the same volume for the polyhedron of possible solutions for unidentifiable links (i.e., $Vol(B) = Vol(R)$), and therefore it provides the same minimum inference error.

[1]Note that, upper and lower bounds for the link delays can be easily added to the system to ensure $Vol(Q) < \infty$.

Algorithm 1: Greedy camera placement algorithm.

```
Input: Matrix R
Q = ∅;
while rank(Q) ≤ rank(R) do
    p* = path in R \ Q with minimum cost and linearly independent from the paths in Q;
    Q = Q ∪ p*;
    foreach Path p ∈ Q \ R do
        Update cost c_p considering the cameras already required by Q
return Q
```

Theorem 3. *Consider a binary matrix R of size* $|P| \times |E|$ *and an end-to-end measurement vector* $\mathbf{b}$*. For any basis B of R, the polyhedron of possible solutions for the system* $R\mathbf{x} = \mathbf{b}$ *is the same of the polyhedron of the solutions of the system* $B\mathbf{x} = \mathbf{b}_B$*.*

Proof. A solution $\mathbf{x}$ for the system $R\mathbf{x} = \mathbf{b}$ exists if the vector $\mathbf{b}$ lies in the span of the vectors in R [22]. Since B is a basis of R, they both span the same vector space. As a result, $\mathbf{x}$ is also a solution for $B\mathbf{x} = \mathbf{b}_B$. □

By combining Theorems 2 and 3 we can conclude that in order to obtain maximum coverage and minimum error while minimizing the cost of deploying cameras, we need to look for the basis of the matrix R with minimum cost. This may be formally stated as an optimization problem, as follows:

$$\begin{aligned} \underset{Q \subseteq R}{\text{minimize}} \quad & C(Q) \\ \text{subject to} \quad & Q \text{ is a basis of } R \end{aligned} \tag{2}$$

where $C(Q)$ is the cost of deploying monitors at the intersections identified by the set of paths Q. The solution Q of the above problem can be easily translated into a set of intersections M by placing cameras at the end points of each path in Q.

We note that $C()$ is a submodular function [18]. Additionally, the linearly independent subsets of R form a matroid. As a result, our problem is the minimization of a submodular function over a matroid constraint. The unconstrained minimization of a submodular function can be solved in polynomial time using the Lovász extension [18].

However, even simple cardinality constraints can make the problem hard [18]. For this reason, in this paper, we propose a greedy algorithm to solve the optimization problem in Eq. (2).

The algorithm takes as input the matrix R obtained by placing cameras at all intersections in V'. It then starts with an empty solution Q (line 1), which is iteratively expanded by the while loop (lines 2-6). The loop iterates as long as Q is not a basis (i.e., its rank is less than that of R). At each iteration, the path with least cost, which is linearly independent from the paths in Q, is selected and added to Q (lines 3-4). The inner for loop (lines 5-6) updates the costs of the paths not in Q by taking into account the cameras required by the paths in Q. This is a necessary step to take into account the submodularity of the objective function.

Complexity Analysis. The while loop runs at most $rank(R)$ times. Calculating the rank can be performed in $O(|E|^3)$ using Gaussian Elimination. Finding the path with minimum cost can be done in $O(|P|)$. Finally, the for loop to update the costs performs $O(|P|)$ iterations for each iteration of the while loop. As a result, the overall complexity is $O(rank(R) \times (|E|^3 + |P|))$.

4.3. SOLVING A LINEAR SYSTEM OF NOISY MEASUREMENTS. In this section, we address two specific challenges arising when applying Network Tomography to vehicular traffic monitoring. First, traffic conditions, drivers' behavior, and other external factors will likely influence the traveling times between any pair of monitored intersections, so we must assume that the corresponding delay measurement will be affected by unpredictable noise. Such noise may result in an inconsistent linear system, with no solution. Second, given a camera placement, there may still be unidentifiable links, and therefore multiple solutions for the linear system. Therefore, in this section, we discuss our approach to deal with noisy measurements and to identify a solution that is consistent with the available information and provides a small inference error of the traveling times for unidentifiable links.

4.3.1. Dealing with Noisy Measurements. Let us consider a given set of cameras $M \subseteq V'$ obtained as discussed in Section 4.2. As previously mentioned, each pair of cameras $m_s, m_d \in M$ identifies a set $P_{s,d}$ of paths; for each path $p \in P_{s,d}$ the Traffic Control Center (TCC) will compute the average traveling time b_p which will be used in the linear system. However, not only might b_p be intrinsically affected by a non-negligible amount of noise due to different cars spending different times to traverse the same path, but cars might as well follow different paths altogether between the same two intersections. As a result, the average b_p for a path p may be affected by significant noise. If the standard network tomography approach were to be used, the linear system in Eq. (1) might have no solution.

To deal with this problem, we modify the original formulation to account for the possible presence of noise in the measurements. Specifically, we define a parameter Δ representing the allowed margin of noise, and substitute each of the equations in the linear system by two inequalities:

$$\sum_{j=1}^{|E|} r_{ij} x_j \leq b_i + \Delta; \quad \sum_{j=1}^{|E|} r_{ij} x_j \geq b_i - \Delta \tag{3}$$

A sufficiently large value for Δ allows making sense of the available end-to-end measurements, so that a solution for the system may exist. However, an excessively large value for this parameter would result in an inaccurate solution, differing significantly from the actual values of the traveling times of the road segments.

Our goal is thus to find the minimum value of Δ that enables a solution of the linear system while also minimizing the error in inferring the road segments traveling times. To this purpose, we define the following linear optimization problem:

$$\begin{aligned} &\text{minimize} && \Delta \\ &\text{subject to} && R\mathbf{x} \leq \mathbf{b} + \Delta,\ R\mathbf{x} \geq \mathbf{b} - \Delta \\ & && \mathbf{x} \geq 0 \end{aligned} \tag{4}$$

where R is the binary matrix of size $|P| \times |E|$. Note that, here the set P accounts for the possibility of multiple paths between any two intersections where cameras are installed.

The problem can be solved efficiently by well-known linear optimization algorithms such as the Simplex method [28], and the solution would provide a value x_j for each road segments $l_j \in E$, that we use as an estimate of the average traveling time for that link if l_j is identifiable.

Note that we may extend the above formulation to include vehicle classes. It may be possible to derive a linear system for each class to obtain more accurate information. However, this may come at the expense of additional noise due to potentially inaccurate vehicle classification, or inherent noise in each class. We will consider vehicle classes as a future extension of our work.

4.3.2. Solution for Unidentifiable Links. It may be the case that only some road segments are identifiable. For such segments the optimizer would return a unique solution; conversely, the values for non-identifiable links would not be unique. In general, the optimizer provides a solution in the form of a point on the surface of the convex polyhedron H defined by the constraints of the optimization problem. We already discussed in Section 4.2 how to place cameras so that such polyhedron is as small as possible. However, unidentifiable links cannot be avoided in general, so we need a method to pick a "good" solution within the polyhedron that provides a small inference error.

The intuition behind our approach is that even though the vertices of the polyhedron may be admissible solutions, they are likely not convenient for us. Such solutions are characterized by very high traveling times assigned to few links, and very low ones to others. Such disproportionate polarization is not realistic, therefore in our approach, the goal is to return a solution which corresponds to the centroid of the polyhedron.

Definition 4 (Centroid)**.** *Given a polyhedron $H \subseteq \mathbb{R}^n$ with vertices $= \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, the centroid $\mathcal{B}(H) \in \mathbb{R}^n$ is calculated as $\mathcal{B}(H) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i$.*

Note that since H is convex, as it results from the intersection of convex polyhedrons defined by the linear inequalities, the centroid $\mathcal{B}(H)$ always lies within H, and therefore it is a feasible solution for the linear system. Calculating the centroid, however, requires the knowledge of all the vertices of the polyhedron. Identifying such vertices is feasible using a similar approach like the one adopted by the Simplex algorithm [28]. However, the number of such vertices can be exponential with the number of constraints. Therefore we propose an alternative approach to provide an approximated centroid.

Our approach consists in generating a set W of random points that lie within the polyhedron H identified by the linear system in Eq. (4). We iteratively generate random points and accumulate in W those that verify the constraints of the linear system. Subsequently, we calculate the centroid of $\mathcal{B}(W)$ as an approximation of the centroid $\mathcal{B}(H)$. We show in Section 5 that this approach provides a very accurate estimation of the traveling times of the links in the road network.

4.4. PAIRING END-TO-END MEASUREMENTS TO PATHS. We now focus on the last challenge and deal with the possible different paths that drivers follow between two monitors. Note that this problem does not occur in the original formulation of network tomography for communication networks since routing is either source-based or known a priori [2].

As an example of this problem, with reference to the linear system in Figure 1, traveling times b_1 and b_2 are both computed from measurements collected by the same pair of monitors (m_1 and m_4); however, b_1 should be associated to path $p_1 = \{l_1, l_7, l_9\}$, whereas b_2 to path $p_2 = \{l_1, l_2, l_8, l_9\}$.

In general, let $P_{s,d}$ indicate the set of all paths that may be potentially traveled by cars having two monitored road intersections $m_s, m_d \in M$ as source and destination, respectively. Paths in $P_{s,d}$ may differ in length and traffic conditions, and consequently, be characterized by significantly different traveling times. Moreover, we do not expect a car to be an active part of our framework; in particular, they are not assumed to be equipped with any tracking

device, such as GPS, that might enable us to infer which of the paths in $P_{s,d}$ was traversed by a specific car. As a consequence, we are faced with the problem of correctly assigning a delay measurement to a specific path.

Formally, the TCC may compute the traveling time t_j for each vehicle j traveling from m_s to m_d. Additionally, in order to build the linear system described in Eq. (4), it needs to compute the average traveling time b_i for each path $p_i \in P_{s,d}$. Let $T_{s,d} = \{t_1, t_2, \ldots, t_r\}$ be the set of end-to-end measurements collected for any of the paths in $P_{s,d}$ between monitors m_s and m_d, and let $|T_{s,d}| = r$ and $|P_{s,d}| = k$.

Our problem consist in determining a partition $T_1, T_2, \ldots, T_k$ of $T_{s,d}$ such that each T_i contains only the end-to-end measurements that belong to path $p_i \in P_{s,d}$. Once the sets $T_1, T_2, \ldots, T_k$ are identified, the average end-to-end traveling time for a path p_i will be computed as $b_i = \frac{1}{|T_i|} \sum_{t \in T_i} t$. This value can be used as the right-hand side for the inequalities related to path p_i in the linear system described in Section 4.3.

In order to find the partitioning of $T_{s,d}$, we assume that the probability $\mathbb{P}_{s,d}(p_i)$ with which a car follows a path p_i between m_s and m_d is known, so that we know that the expected number of measurements belonging to set T_i is given by $r \times \mathbb{P}_{s,d}(p_i)$, where we recall $r = |T_{s,d}|$. Note that, in Section 5.5 we relax this assumption and experimentally show that our approach is effective even when there is an inaccurate knowledge of the probability distribution.

We model the traveling times along a path p_i by means of a random variable X_i, whose probability distribution is not restricted to have any specific shape. T_i is then the set of realizations of such random variable. If $\mathbb{P}(X_i = t)$ indicates the probability that a sample $t \in T_{s,d}$ has been generated by X_i, the *likelihood* of an assignment $T_1, \ldots, T_k$ is given by:

$$\mathbb{P}(T_1, \ldots, T_k) = \Pi_{i=1}^{k} \Pi_{t \in T_i} \mathbb{P}(X_i = t) \tag{5}$$

Algorithm 2: Assignment of end-to-end measures to paths between a pair of monitors.

```
1  Generate a random assignment T_1, T_2, ..., T_k;
2  Calculate the empirical distributions P̂_i for each X_i;
3  do
4      δ* = 0;
5      foreach pair of samples t_1, t_2 ∈ T_{s,d} do
6          Calculate δ_{1,2} as the increase in P(T_1, ..., T_k) by switching t_1 and t_2;
7          if δ_{1,2} > 0 and δ* < δ_{1,2} then
8              δ* = δ_{1,2}
9      Switch t_1 and t_2 that provided δ*
10 while δ* > δ_min;
11 return the current assignment T_1, T_2, ..., T_k
```

and the best assignment $T_1, \ldots, T_k$ is one that maximizes such likelihood. In the ideal case when the distributions are known for each variable X_i, the best partitioning can be determined by assigning each measurement t to the variable X_i for which $\mathbb{P}(X_i = t)$ is the highest.

In our case, however, the distributions are not known, so we resort to an iterative greedy approach inspired by the Kernighan-Lin graph partitioning algorithm [17]. The idea is to start from a random assignment respecting the cardinality constraints for each set T_i, and iteratively make use of the samples in T_i to estimate the *empirical distribution* $\hat{\mathbb{P}}_i$ of each variable X_i. Then, for each pair of samples $t_1 \in T_i$ and $t_2 \in T_j$ we calculate $\delta_{1,2}$ which is defined as the increase in the likelihood of the assignment in Eq. (5) by swapping t_1 and t_2, i.e. by assigning t_1 to T_j and t_2 to T_i. The pair of samples with highest δ are then switched at the current iteration. The algorithm terminates as soon as the increase drops below a threshold δ_{min}.

The pseudo-code for the algorithm is in shown in Algorithm 2. Each iteration of the algorithm has $O(r^3)$ complexity, since each pair of samples $t_1, t_2 \in T_{s,d}$ must be analyzed, and, for each of these pairs, $\delta_{1,2}$ must be computed, an operation whose complexity is upper bounded by $O(r)$.

In order to ensure convergence of the algorithm, $\delta_{1,2}$ is calculated as follows. For $t_1 \in T_i$ and $t_2 \in T_j$, we simulate the swapping of t_1 and t_2, obtaining the corresponding sets $T_i' = T_i \setminus \{t_1\} \cup \{t_2\}$ and $T_j' = T_j \setminus \{t_2\} \cup \{t_1\}$ and the new empirical distributions $\hat{\mathbb{P}}_i'$ and $\hat{\mathbb{P}}_j'$, for X_i and X_j. $\delta_{1,2}$ is defined as:

$$\begin{aligned} \delta_{1,2} =& max\{(\Pi_{t \in T_i'} \hat{\mathbb{P}}_i'(X_i = t) - \Pi_{t \in T_i} \hat{\mathbb{P}}_i(X_i = t)), 0\} \times \\ & max\{(\Pi_{t \in T_j'} \hat{\mathbb{P}}_j'(X_j = t) - \Pi_{t \in T_j} \hat{\mathbb{P}}_j(X_j = t)), 0\} \end{aligned} \tag{6}$$

The rationale behind the above formula is that whenever t_1 and t_2 are exchanged, a strictly positive increase in the probability of the new assignment occurs even if the empirical distributions of X_i and X_j may change.

5. EXPERIMENTAL ASSESSMENT

5.1. EXPERIMENTAL SETUP. In this section, we experimentally evaluate our framework through simulations. We consider both synthetic and real road networks. Synthetic road networks have been generated using the realistic model proposed in [11]. The model is based on quadtrees and has two parameters: the number of intersections f, and q which controls the amount of sprawl. In our experiments, f is varied to consider different scenarios, and q is set to 1. Traveling times are randomly generated in the interval $[3, 10]$, which represents the ground truth for our inference. An example of the topology generated with this model is shown in Figure 2.

The real road network topology is instead taken from downtown San Francisco, CA, as shown in in Figure 3. When required, we generate networks of varying size by considering only the part of the road network comprised within a specified radial distance from a geographical center of the map. We developed a script based on the Google Maps API [12] to obtain real traveling times. Specifically, our script uses the APIs to obtain the current traveling time in every road segment in the downtown San Francisco map.

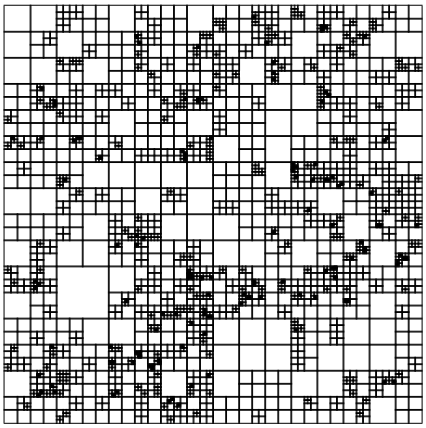


Figure 2. Example of quadtree based synthetic topology [11].

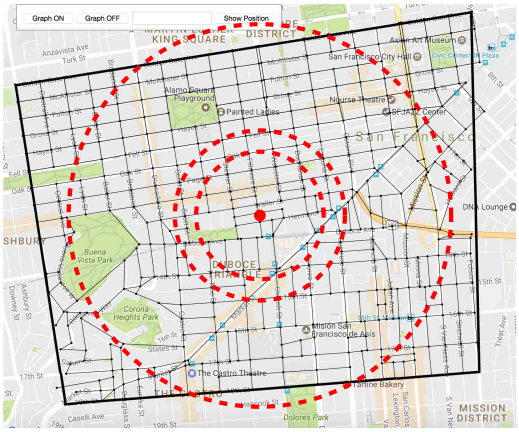


Figure 3. Road topology used for the experiments.

Figure 4 shows the Cumulative Density Function (CDF) for the observed delays at 2PM on 8/8/2017[2]. We used these values as ground truth for our experiments with the real road network topology.

For both synthetic and real networks, we generated the cost c_m of deploying a camera at intersection $m \in V'$ as a random number in the interval $[1, 10]$.

In order to account for the fact that in a real-life scenario, drivers may decide to follow alternate, but not entirely dissimilar, routes for each pair of intersections m_s, m_d, we consider all paths in the set $P_{s,d}$ whose number of hops is less than or equal to a factor θ of

[2]We also considered different times and days, and we obtained similar results.

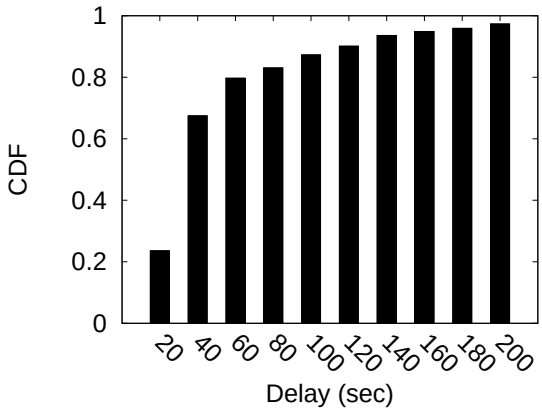


Figure 4. Cumulative Density Function (CDF) of delays observed in downtown San Francisco, CA.

the length of the shortest path between m_s and m_d. The set $P_{s,d}$ was constructed through a modified version of the transitive closure of the adjacency matrix of the roadmap graph G [29]. In our experiment, we show the impact of the parameter θ on the performance of our approach. We also assumed that collected end-to-end traveling times may be randomly corrupted by some amount of noise, and we performed experiments to show the impact of such noise.

We focus on three main metrics to evaluate our approach. Cost refers to the total cost of deploying cameras at the selected intersections. Coverage is defined as the fraction of links for which an estimation is provided with respect to the total number of links in E. The inference error is measured using the Mean Square Error (MSE). Specifically, let $x_1, \ldots, x_{|E|}$ be the ground truth values of delays in the network, and let $\hat{x}_1, \ldots, \hat{x}_{|E|}$ be the inferred values. The MSE is defined as $MSE = \frac{1}{|E|} \sum_{i=1}^{|E|} (\hat{x}_i - x_i)^2$. Note that, for uncovered links, we assume that the inferred delay is zero so that the corresponding error is maximized. The availability of additional information is available, e.g., road lengths and speed limits would result in better inference.

Results are averaged over several runs to obtain a reliable confidence interval. The plots show the average and standard deviation of the considered metrics.

5.2. A COMPARISON APPROACH BASED ON VERTEX COVER. Previous works on vehicular traffic estimation based on cameras [14, 27, 31] do not address the camera placement problem. Similarly, monitor placement in communication networks is based on routing assumptions that do not hold for vehicular networks [19, 23, 24, 35]. For these reasons, we chose instead to compare our method to an approach inspired by the Weighted Vertex Cover (WVC) problem [6], which has been adopted in several coverage problems in communication and vehicular networks [20]. Specifically, given the road network $G = (V, E)$ and the costs c_m for each $m \in V$, we assume that if a camera is placed at an intersection $m \in V$, the traffic along all the links in E incident to m can be accurately monitored. In terms of the WVC problem, we say that m covers the links adjacent to it. Under this setting, we are looking for a set $M_{WVC} \subseteq V$ to place cameras such that it incurs minimum costs and each edge in E is covered by at least one camera in M_{WVC}. A similar formulation of the problem is straightforward for the case when cameras are allowed to be placed only at a subset $V' \subseteq V$ of intersections.

The WVC problem is NP-Complete, however, there are several heuristics that provide a provable approximation bound. In this paper we use the heuristic proposed by Clarkson et al. that provides a 2 approximation bound with respect to the optimal solution [6]. This approach is referred to as WVC.

5.3. SYNTHETIC NETWORKS. Experiment I. In the first set of experiments we allow cameras to be potentially deployed at any intersection in the network, i.e., $V' = V$. In addition, we consider three values for θ, namely 1, 1.2, and 1.5, and we consider no errors in the measurements nor the assignment of measurements to paths.

Figure 5(a) shows the total cost of deploying cameras at intersections returned by our approach, and at those selected by WVC, under different network sizes. Our approach outperforms WVC. The strength of our solution is the ability to exploit the correlation between end-to-end measurements through the solution of the linear system. Conversely, each camera in WVC is only able to provide a local estimation, although unbiased, of its

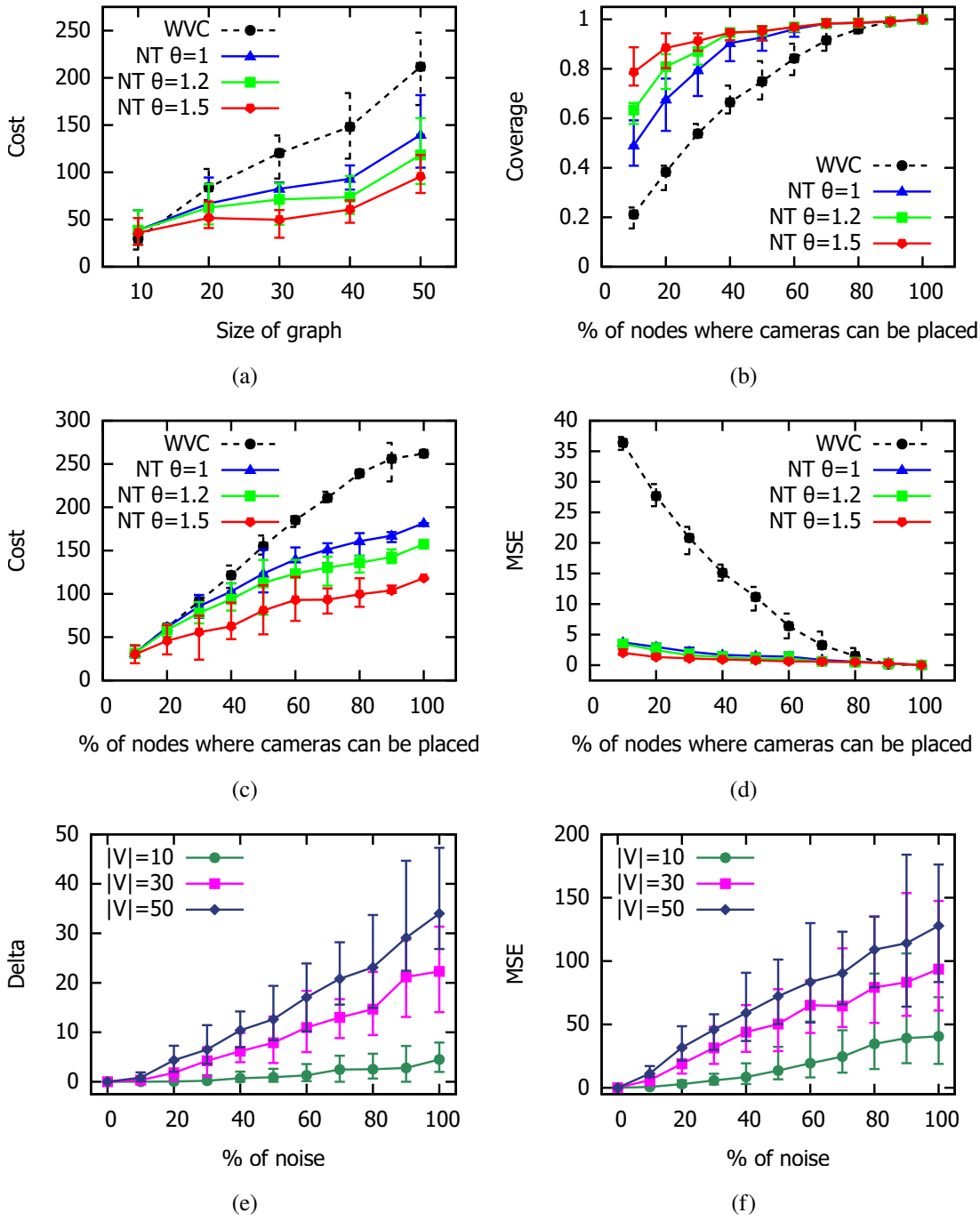


Figure 5. Synthetic Networks: Cost vs. size of the network (a), Coverage (b), Cost (c) and MSE (d) vs. % of nodes where cameras can be placed, Δ (e), MSE (f) vs. % of noise.

adjacent links. As a result, WVC requires more cameras, and therefore it incurs a higher cost. In addition, our solution requires a lower cost as the value of θ increases, since fewer cameras are necessary to cover the network. Nevertheless, even in the extreme case of a single path between each pair of cameras ($\theta = 1$), we still significantly outperform WVC.

Note that, by allowing cameras to be potentially deployed at every intersection, our approach can uniquely identify all links in the network. Similarly, WVC returns a set of nodes for which every link is adjacent to at least a node in that set. As a result, both approaches achieve full coverage and incur a null inference error.

Experiment II. In this set of experiments, we allow cameras to be placed only at a subset of intersection $V' \subseteq V$, and we study the performance upon increasing the size of V'. The intersections in V' are chosen randomly, and again we assume no errors are present.

Figure 5(b) and (c) show the coverage and the cost of camera deployment versus the size of V', expressed as the percentage of nodes in V where cameras can be placed. Remarkably, our approach is able to provide higher coverage at a smaller cost compared to WTC by exploiting end-to-end measurements. As a numerical example, our approach provides a coverage higher than 90% coverage of the network, with $\theta = 1.2$, when having only 30% of intersections in V' and incurring a cost less than 75. On the contrary, WTC requires at least 70% of intersection available to achieve a similar coverage, and it incurs a cost of about 150.

The lower coverage also has an impact on the quality of the inference, as shown in Figure 5(d). Our approach incurs a very low error for all settings of θ, again highlighting the inference capabilities of network tomography when applied to vehicular networks. Conversely, WTC results in an error which is significantly higher than our approach.

Experiment III. In the third and final set of experiments, we focus on the effect of noisy measurements. For this purpose, we consider the end-to-end measurements to be affected by a random noise $e \in [0, 1]$, and we study the performance by increasing the value of e. Specifically, depending on e we alter the ground truth b_i value for the delay of path p_i in

Eq. 4, with a random value in the interval $[(1-e)\times b_i, (1+e)\times b_i]$. This makes the linear system likely to have no solution. Therefore a value of $\Delta > 0$ is necessary to satisfy all the constraints given the noisy measurements. In these experiments we allow cameras to be potentially deployed at all of the intersections.

Figure 5(e) shows the values of Δ corresponding to increasing values for the error parameter for network topologies of size 10, 30, and 50 nodes. As expected, a higher value of Δ is required as the noise increases, and also a higher value is necessary to satisfy the constraints for bigger topologies. Intuitively, the bigger the topology, the higher the number of paths, the higher the impact of noise on the feasibility of the linear system. Figure 5 (f) shows the inference error under different magnitude of noise. The error increases with the amount of noise and the size of the network. This increase is due to the higher values of Δ, that result in larger polyhedrons, and consequently to larger inference errors. Nevertheless, it is worth noting that although the error increases, our approach is effectively able to achieve a small inference error in all the considered settings.

5.4. REAL NETWORKS. We now describe the results obtained using the road network of downtown San Francisco, CA. We performed similar experiments with respect to the synthetic networks.

Experiment I. Figure 6(a) shows the cost of deploying cameras at selected intersections as a function of the network size. The results confirm the outcome obtained for synthetic networks, and also, in this case, our approach outperforms WTC incurring a lower cost. The advantage becomes more evident as the network size increases since the effect of exploiting end-to-end measurements is more prominent.

Experiment II. We now consider the performance when increasing the size of the set V' where cameras can be placed. As figures 6(b), (c) and (d) show, WVC reveals its weaknesses in real settings too, achieving lower coverage, at a higher cost, which results in a higher inference error for the link delays. Note that the higher values of the MSE compared to the

synthetic networks are due to the higher delays of the links in this dataset, in the interval $[10, 300]$ as obtained by the script based on the Google APIs. Conversely, the delays for the synthetic networks are in the interval $[3, 10]$.

Experiment III. In the final set of experiments, we added noise to the measurements as discussed for the case of synthetic networks. Figure 6(e) and (f) show the value of Δ and the inference error as a function of the magnitude of added noise. Similarly to the synthetic network case, a higher Δ and a higher error occur as we increase the noise and the size of the network. Nevertheless, even in this case, our approach shows great tolerance to noisy measurements, as the MSE, which is a quadratic error, increases only linearly with the noise.

5.5. ASSIGNING MEASUREMENTS TO PATHS. In this set of experiments, we aim to evaluate the performance of the algorithm presented in Section 4.4 whose goal is to assign the measurements $T_{s,d}$ collected between two intersections m_s and m_d, to the paths $P_{s,d}$ between them. We assume that each path $p_i \in P_{s,d}$ has a ground truth average delay μ_i and that the traveling times over p_i follow a Gaussian distribution $< \mu_i, \sigma >$. The value of μ_i is picked randomly in the interval [1,100], while σ is set to 3. We additionally assume that the driver preference for the available paths follows a geometric distribution with parameter $\lambda \in [0, 1]$, meaning that path $p_i \in P_{s,d}$ is picked with probability $(1 - \lambda)^i$. This distribution models the real-life scenario in which most drivers tend to concentrate over few paths.

Using these assumptions, we simulate 800 cars traveling from m_s to m_d. Each car picks a path p_i in $P_{s,d}$ according to the geometric distribution. Then, we add a measurement in the set $T_{s,d}$ as realization of the Gaussian distributed random variable with parameters $< \mu_i, \sigma >$.

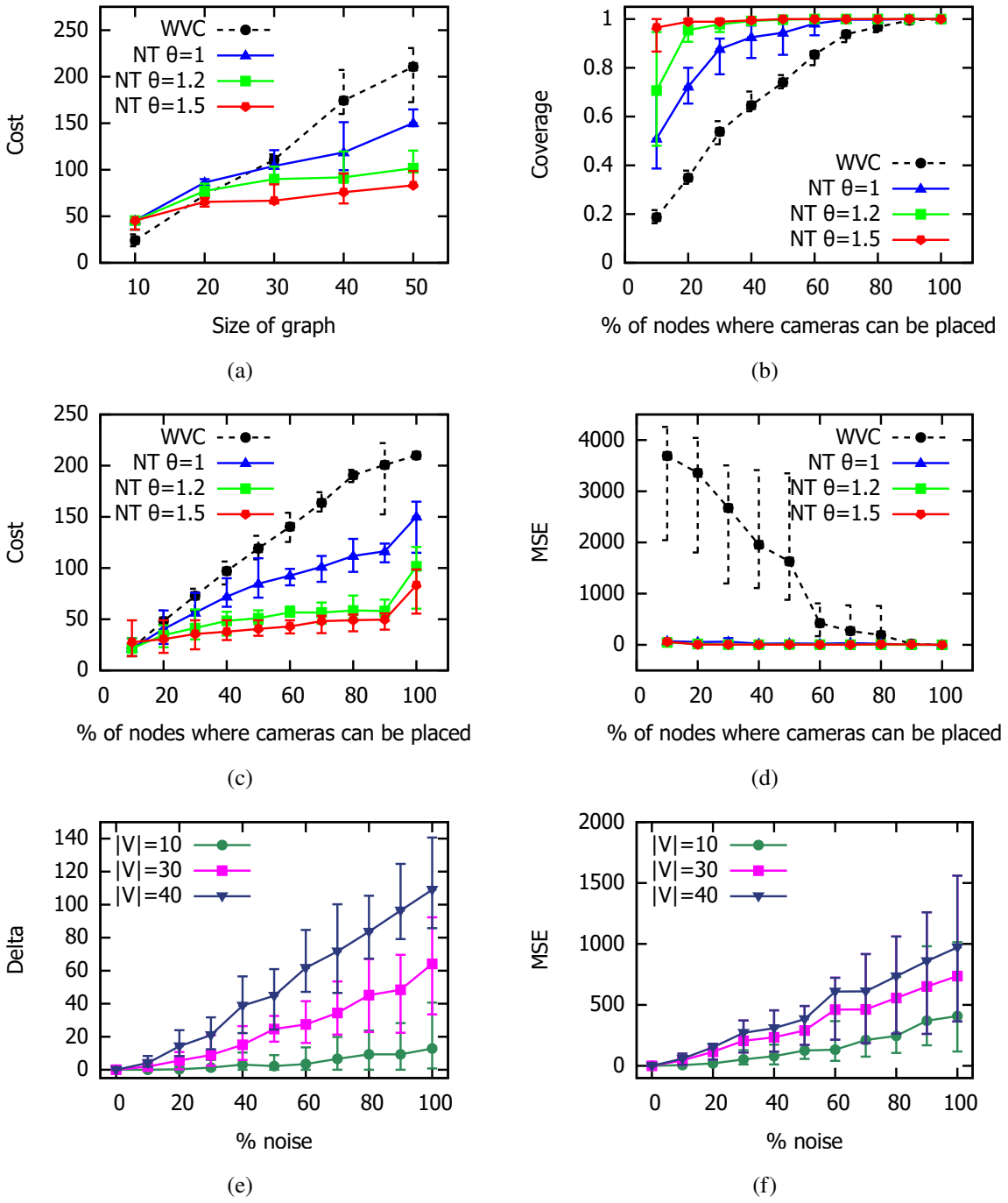


Figure 6. Real Networks: Cost vs. size of the network (a), Coverage (b), MSE (c) and Cost (d) vs. % of nodes where cameras can be placed, Δ (e), MSE (f) vs. % of noise.

We run our algorithm to partition the set $T_{s,d}$ into $T_1, T_2, \ldots$, where T_i contains the measurements assigned to the path p_i. The estimated average traveling time for p_i is $\hat{\mu}_i$, obtained averaging the measurements in T_i. We use the Mean Square Error (MSE) between the actual averages and the estimated averages to evaluate the accuracy of our approach, specifically $MSE = \frac{1}{|P_{s,d}|} \sum_{p_i \in P_{s,d}} (\hat{\mu}_i - \mu_i)^2$.

We compare our approach with the clustering algorithm K-means [25]. This algorithm starts from K randomly assigned centroids, where for us $K = |P_{s,d}|$, and it assigns the measurement in $T_{s,d}$ to the closest centroid. At each subsequent iteration, centroids are updated based on the previous measurement assignment, and measurements are assigned again to the closest updated centroid. The process is repeated until a convergence criterion is met. We refer the reader to [25] for more details about K-means.

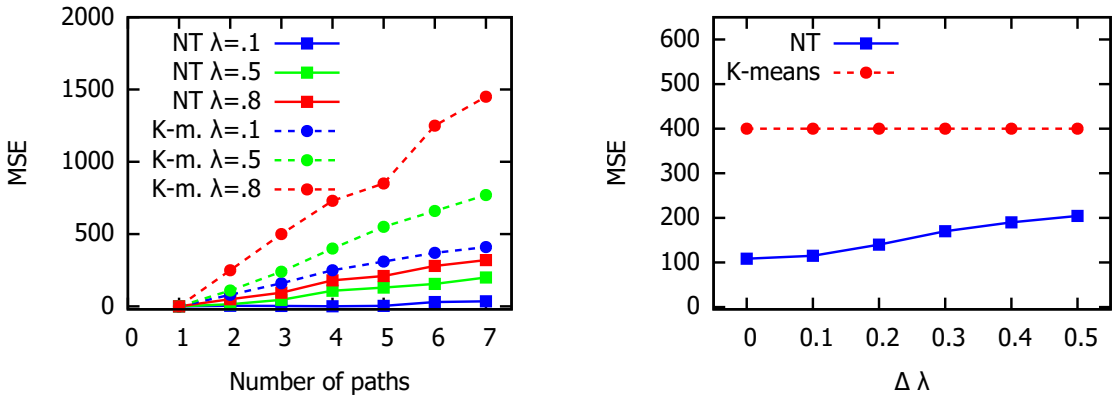


Figure 7. Assigning measurements to paths, with (a) perfect and (b) inaccurate knowledge of the driver preference distribution.

We initially assume that the parameter λ of the geometric distribution of the driver preference is known. Figure 7(a) shows the results increasing the number of paths in $T_{s,d}$. As expected, the availability of a larger number of paths makes the problem harder. Nevertheless, our approach incurs in a small error, and it outperforms K-Means by providing a more accurate estimation of the traveling times in all settings.

Next, we relax the assumption on perfect knowledge of the driver preference distribution. Specifically, we set $\lambda = 0.5$ but run the algorithm assuming a distribution with parameter $\lambda + \Delta\lambda$, where $\Delta\lambda$ represents the amount of knowledge inaccuracy. Figure 7 (b) shows the MSE for 4 paths increasing the value of $\Delta\lambda$. K-Means is not affected by $\Delta\lambda$, since it does not make use of the underlying driver preference distribution. Our approach shows very high robustness as it is able to provide better accuracy than K-means, even under severe inaccurate knowledge.

6. CONCLUSIONS

This paper addressed the issue of vehicular traffic monitoring in a smart city. We described a method for inferring the traveling times on each road segment of the city while requiring just a minimum number of monitoring devices to be deployed at selected intersections. The theoretical grounds for our method are provided by the Network Tomography approach. We formulated an optimization problem for the optimal placement of monitoring cameras and exploited linear algebra to propose an efficient greedy solution. We additionally adapted the classical network tomography approach by allowing for noisy measurements and unpredictability of vehicles paths. Experimental results on real and synthetic networks show that our approach provides a cost-efficient deployment of cameras that allows achieving full coverage of the road network and a very low inference error.

REFERENCES

[1] Atherton, K., *Israeli students spoof Waze with fake traffic jam*, 2014, `http://tinyurl.com/p4gcgkv`.

[2] Bartolini, N., He, T., and Khamfroush, H., 'Fundamental limits of failure identifiability by boolean network tomography,' in 'IEEE INFOCOM,' 2017 .

[3] Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., and Portugali, Y., 'Smart cities of the future,' The European Physical Journal Special Topics, 2012, **214**(1), pp. 481–518.

[4] Bu, T., Duffield, N., Lo Presti, F., and Towsley, D., 'Network tomography on general topologies,' in 'ACM SIGMETRICS Performance Evaluation Review,' volume 30, ACM, 2002 pp. 21–30.

[5] Chen, Y., Bindel, D., Song, H., and Katz, R. H., 'An algebraic approach to practical and scalable overlay network monitoring,' ACM SIGCOMM Comp. Com. Rev., 2004, **34**(4), pp. 55–66.

[6] Clarkson, K. L., 'A modification of the greedy algorithm for vertex cover,' Information Processing Letters, 1983, **16**(1), pp. 23–25.

[7] Datondji, S. R. E., Dupuis, Y., Subirats, P., and Vasseur, P., 'A survey of vision-based traffic monitoring of road intersections,' IEEE Transactions on Intelligent Transportation Systems, 2016, **17**(10), pp. 2681–2698.

[8] Denaxas, E., Mpollas, S., Vitsios, D., Zolotas, C., Bleris, D. G., Spanos, G. M., and Pitsianis, N. P., 'Real-time urban traffic information extraction from GPS tracking of a bus fleet,' in 'IEEE CIVTS,' 2013 .

[9] Du, R., Chen, C., Yang, B., Lu, N., Guan, X., and Shen, X., 'Effective urban traffic monitoring by vehicular sensor networks,' IEEE Transactions on Vehicular Technology, 2015, **64**(1), pp. 273–286.

[10] Du, S., Ibrahim, M., Shehata, M., and Badawy, W., 'Automatic license plate recognition (alpr): A state-of-the-art review,' IEEE Transactions on Circuits and Systems for Video Technology, 2013, **23**(2), pp. 311–325.

[11] Eisenstat, D., 'Random road networks: the quadtree model,' in 'SIAM ANALCO,' 2011 .

[12] Google, *Google Maps APIs*, 2017, `https://developers.google.com/maps/`.

[13] Gopalan, A. and Ramasubramanian, S., 'On identifying additive link metrics using linearly independent cycles and paths,' IEEE/ACM Transactions on Networking (TON), 2012, **20**(3), pp. 906–916.

[14] Idé, T., Katsuki, T., Morimura, T., and Morris, R., 'City-wide traffic flow estimation from a limited number of low-quality cameras,' IEEE Transactions on Intelligent Transportation Systems, 2017, **18**(4), pp. 950–959.

[15] Idé, T. and Sugiyama, M., 'Trajectory regression on road networks.' in 'AAAI,' 2011 .

[16] Kanistras, K., Martins, G., Rutherford, M. J., and Valavanis, K. P., 'Survey of unmanned aerial vehicles (uavs) for traffic monitoring,' in 'Handbook of Unmanned Aerial Vehicles,' pp. 2643–2666, Springer, 2015.

[17] Kernighan, B. W. and Lin, S., 'An efficient heuristic procedure for partitioning graphs,' The Bell system technical journal, 1970, **49**(2), pp. 291–307.

[18] Krause, A. and Golovin, D., 'Submodular function maximization.' 2014.

[19] Kumar, R. and Kaur, J., 'Practical beacon placement for link monitoring using network tomography,' IEEE Selected Areas in Communications, 2006, **24**(12), pp. 2196–2209.

[20] Lam, A. Y., Leung, Y.-W., and Chu, X., 'Electric vehicle charging station placement: Formulation, complexity, and solutions,' IEEE Transactions on Smart Grid, 2014, **5**(6), pp. 2846–2856.

[21] Lawrence, E., Michailidis, G., Nair, V. N., and Xi, B., 'Network tomography: A review and recent developments,' in 'In Fan and Koul, editors, Frontiers in Statistics,' College Press, 2006 pp. 345–364.

[22] Lay, D., *Linear Algebra and Its Applications*, Pearson Education, 2002, ISBN 9788177583335.

[23] Ma, L., He, T., Leung, K. K., Swami, A., and Towsley, D., 'Inferring link metrics from end-to-end path measurements: Identifiability and monitor placement,' IEEE/ACM Transactions on Networking (TON), 2014, **22**(4), pp. 1351–1368.

[24] Ma, L., He, T., Swami, A., Towsley, D., and Leung, K. K., 'On optimal monitor placement for localizing node failures via network tomography,' Performance Evaluation, 2015, **91**, pp. 16–37.

[25] MacKay, D. J., *Information theory, inference and learning algorithms*, Cambridge university press, 2003.

[26] Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., and Diot, C., 'Traffic matrix estimation: Existing techniques and new directions,' ACM SIGCOMM Computer Communication Review, 2002, **32**(4), pp. 161–174.

[27] Morimura, T., Osogami, T., and Idé, T., 'Solving inverse problem of markov chain with partial observations,' in 'Advances in Neural Information Processing Systems,' 2013 pp. 1655–1663.

[28] Nelder, J. A. and Mead, R., 'A simplex method for function minimization,' The Computer Journal, 1965, **7**(4), pp. 308–313.

[29] Newman, M., *Networks: an introduction*, Oxford university press, 2010.

[30] Oh, S., Ritchie, S., and Oh, C., 'Real-time traffic measurement from single loop inductive signatures,' Transportation Research Record: Journal of the Transportation Research Board, 2002, (1804), pp. 98–106.

[31] Santini, S., 'Analysis of traffic flow in urban areas using web cameras,' in 'IEEE WACV,' 2000 pp. 140–145.

[32] Smart Cities Council, 'Smart cities readiness guide: The planning manual for building tomorrow's cities today,' Smart Cities Council, Redmond, 2014.

[33] Sohn, K. and Hwang, K., 'Space-based passing time estimation on a freeway using cell phones as traffic probes,' IEEE Transactions on Intelligent Transportation Systems, 2008, **9**(3), pp. 559–568.

[34] Tati, S., Silvestri, S., He, T., and La Porta, T., 'Robust network tomography in the presence of failures,' in 'IEEE ICDCS,' 2014 .

[35] Zad, D., He, T., and La Porta, T., 'Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring,' IFIP Performance, 2017.

[36] Zheng, Q. and Cao, G., 'Minimizing probing cost and achieving identifiability in probe-based network link monitoring,' IEEE Transactions on Computers, 2011, **62**(3), pp. 510–523.

SECTION

2. SUMMARY AND CONCLUSIONS

This thesis discussed a traffic monitoring approach which benefits urban planners of smart cities. In summary, an optimal camera placement strategy is addressed and tested on real and synthetic road network topologies. It is done by not only applying the concept of network tomography but also including noise handling and measurements assigning as complementary. The approach is made to fit challenges existed in road networks and vehicle probing.

Specifically, linear algebra tools, such as linear programming, has been used to define an optimization problem that identifies the set of intersections that would provide maximum coverage, minimum inference error, and minimum cost. The problem is analyzed as an instance of the minimization of a sub-modular function over a matroid constraint, and a greedy algorithm is defined to solve it. The experimental results are compared with a weighted vertex cover approach regarding coverage, error, and cost. In the measurements assigning algorithm, each measurement is matched to its potential path according to perfect or inaccurate knowledge on driver preference distributions. This approach outperforms K-means in all performed experiments.

REFERENCES

[1] Bu, T., Duffield, N., Lo Presti, F., and Towsley, D., 'Network tomography on general topologies,' in 'ACM SIGMETRICS Performance Evaluation Review,' volume 30, ACM, 2002 pp. 21–30.

[2] Clarkson, K. L., 'A modification of the greedy algorithm for vertex cover,' Information Processing Letters, 1983, **16**(1), pp. 23–25.

[3] Kernighan, B. W. and Lin, S., 'An efficient heuristic procedure for partitioning graphs,' The Bell system technical journal, 1970, **49**(2), pp. 291–307.

VITA

Ruoxi Zhang was born and raised in Baoding city, which is in Hebei province of China. In May 2016, Ruoxi received her Bachelor's degree in Computer Science and Mathematics in Southwest Baptist University at Bolivar, Missouri. After that, she joined the computer science department as a master's student in August 2016, and Dr. Simone Silvestri was her academic advisor. Ruoxi also worked with him as a research assistant during her time at Missouri University of Science and Technology. Ruoxi received her master's of science degree with a thesis in Computer Science from Missouri University of Science and Technology in May 2018.